

Docket No.: 22040-00019-US  
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Design Application of:  
Yukio Koyanagi

Application No.: 10/655,352

Confirmation No.: N/A

Filed: September 5, 2003

Art Unit: N/A

For: COMPRESSING DEVICE AND METHOD,  
DECOMPRESSING DEVICE AND METHOD,  
COMPRESSING/DECOMPRESSING  
SYSTEM, PROGRAM, RECORD MEDIUM

Examiner: Not Yet Assigned

**CLAIM FOR PRIORITY AND SUBMISSION OF DOCUMENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

Applicant hereby claims priority under 35 U.S.C. 119 based on the following prior foreign application filed in the following foreign country on the date indicated:

<u>Country</u>	<u>Application No.</u>	<u>Date</u>
Japan	2001-173884	June 8, 2001

In support of this claim, a certified copy of the said original foreign application is filed herewith.

Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 22-0185, under Order No. 22040-00019-US from which the undersigned is authorized to draw.

Dated: October 24, 2003  
13072\_1

Respectfully submitted,

By Larry J. Hume

Larry J. Hume, Registration No.: 44,163  
CONNOLLY BOVE LODGE & HUTZ LLP  
1990 M Street, N.W., Suite 800  
Washington, DC 20036-3425  
(202) 331-7111  
(202) 293-6229 (Fax)  
Attorney for Applicant

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2001年 6月 8日

出願番号  
Application Number: 特願2001-173884  
[ST. 10/C]: [JP2001-173884]

出願人  
Applicant(s): 有限会社ニューロソリューション

2003年10月 2日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫

出証番号 出証特2003-3081536

【書類名】 特許願

【整理番号】 13NS1324

【提出日】 平成13年 6月 8日

【あて先】 特許庁長官殿

【国際特許分類】 H03M 1/00

【発明者】

    【住所又は居所】 埼玉県さいたま市中尾 4 0 9 - 1 - D 1 1 5

    【氏名】 小柳 裕喜生

【特許出願人】

    【識別番号】 595016543

    【氏名又は名称】 酒井 康江

【代理人】

    【識別番号】 100105784

    【弁理士】

    【氏名又は名称】 橘 和之

    【電話番号】 0492-49-5122

【手数料の表示】

    【予納台帳番号】 070162

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 0013545

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 圧縮装置及び方法、伸長装置及び方法、圧縮伸長システム、プログラム、記録媒体

【特許請求の範囲】

【請求項 1】 順次入力されるサンプリングデータを順次遅延させる数段の遅延回路と、

上記数段の遅延回路の各出力段から取り出されるそれぞれのデータに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める乗加算回路とを備えたことを特徴とする圧縮装置。

【請求項 2】 順次入力されるサンプリングデータを順次遅延させる 4 段の遅延回路と、

上記 4 段の遅延回路の各出力段から取り出されるそれぞれのデータに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める乗加算回路とを備えたことを特徴とする圧縮装置。

【請求項 3】 上記乗加算回路は、2 段目および 3 段目の遅延回路の出力データを加算して 9 倍する第 1 の乗加算回路と、

1 段目および 4 段目の遅延回路の出力データを加算して  $-1$  倍する第 2 の乗加算回路と、

上記第 1 の乗加算回路の出力データおよび上記第 2 の乗加算回路の出力データを加算して  $1/16$  倍する第 3 の乗加算回路とを備えることを特徴とする請求項 2 に記載の圧縮装置。

【請求項 4】 上記 4 段の遅延回路と上記乗加算回路とを 1 組の間引き回路として、複数の間引き回路をカスケード接続したことを特徴とする請求項 2 に記載の圧縮装置。

【請求項 5】 順次入力されるサンプリングデータを順次遅延させる 7 段の遅延回路と、

上記 7 段の遅延回路の各出力段から取り出されるそれぞれのデータに対して、



デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める乗加算回路とを備えたことを特徴とする圧縮装置。

【請求項 6】 上記乗加算回路は、1 段目および 7 段目の遅延回路の出力データを加算して  $-1$  倍する第 1 の乗加算回路と、

3 段目および 5 段目の遅延回路の出力データを加算して 9 倍する第 2 の乗加算回路と、

4 段目の遅延回路の出力データを 16 倍する乗算回路と、

上記第 1 の乗加算回路の出力データ、上記第 2 の乗加算回路の出力データ、上記乗算回路の出力データを加算して  $1/32$  倍する第 3 の乗加算回路とを備えることを特徴とする請求項 5 に記載の圧縮装置。

【請求項 7】 上記 7 段の遅延回路と上記乗加算回路とを 1 組の間引き回路として、複数の間引き回路をカスケード接続したことを特徴とする請求項 5 に記載の圧縮装置。

【請求項 8】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるようにしたことを特徴とする圧縮装置。

【請求項 9】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引き手段と、

上記間引き手段により求められた間引きデータを用いて、2 つの間引きデータ間を結ぶ直線上のデータ値と、その直線上のデータ値と同じサンプル点における間引きデータ値との誤差が所望の値以下となるサンプル点を標本点として検出する標本点検出手段と、

上記標本点検出手段により検出された各標本点における離散的な振幅データと

、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得る圧縮データ生成手段とを備えることを特徴とする圧縮装置。

【請求項 1 0】 上記圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換手段を備え、上記置換手段の出力データを上記間引き手段に導出するようにしたことを特徴とする請求項 9 に記載の圧縮装置。

【請求項 1 1】 上記圧縮対象として入力されるサンプリングデータの絶対値を所定値だけ小さく丸めるとともに、絶対値が上記所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換手段を備え、上記置換手段の出力データを上記間引き手段に導出するようにしたことを特徴とする請求項 9 に記載の圧縮装置。

【請求項 1 2】 上記圧縮データ生成手段により求められた上記各標本点における振幅データの下位数ビットを丸める演算を行う丸め演算手段を備え、

上記丸め演算の施された上記各標本点における振幅データと、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得るようにしたことを特徴とする請求項 9 ～ 1 1 の何れか 1 項に記載の圧縮装置。

【請求項 1 3】 上記丸め演算は、丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 1 2 に記載の圧縮装置。

【請求項 1 4】 上記非線形な関係となる演算は、対数関数もしくはそれに近似した関数に基づく演算であることを特徴とする請求項 1 3 に記載の圧縮装置。

【請求項 1 5】 圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換手段と、

上記置換手段より順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引き手段とを備えることを特徴とする圧縮装置。

【請求項 1 6】 上記間引き手段により求められた間引きデータの下位数ビットを丸める演算を行う丸め演算手段を備えることを特徴とする請求項 1 5 に記載の圧縮装置。

【請求項 1 7】 上記丸め演算は、丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 1 6 に記載の圧縮装置。

【請求項 1 8】 上記丸め演算手段により求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮手段を備えることを特徴とする請求項 1 7 に記載の圧縮装置。

【請求項 1 9】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引き手段と、

上記間引き手段により求められた間引きデータの下位数ビットを丸める演算を行う丸め演算手段とを備えることを特徴とする圧縮装置。

【請求項 2 0】 上記丸め演算は、丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 1 9 に記載の圧縮装置。

【請求項 2 1】 上記丸め演算手段により求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮手段を備えることを特徴とする請求項 2 0 に記載の圧縮装置。

【請求項 2 2】 圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換手段と、

上記置換手段より出力されるサンプリングデータに対して、丸め演算前後のデータ値が非線形な関係となる演算によって下位数ビットを丸める演算を行う丸め演算手段と、

上記丸め演算手段により求められたデータについて、絶対値ゼロのデータが所

定数以上続いたときに、それを－0の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮手段を備えることを特徴とする圧縮装置。

【請求項 2 3】 圧縮対象として入力されるサンプリングデータに対して、丸め演算前後のデータ値が非線形な関係となる演算によって下位数ビットを丸める演算を行う丸め演算手段と、

上記丸め演算手段により求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを－0の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮手段を備えることを特徴とする圧縮装置。

【請求項 2 4】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるようにしたことを特徴とする圧縮方法。

【請求項 2 5】 上記デジタル基本関数は、1クロック毎にデータ値が－1，1，8，8，1，－1と変化する関数であることを特徴とする請求項 2 4 に記載の圧縮方法。

【請求項 2 6】 上記デジタル基本関数は、請求項 2 5 に記載のデジタル基本関数を1クロックずつずらして加算することによって生成した、データ値が1クロック毎に－1，0，9，16，9，0，－1と変化する関数であることを特徴とする請求項 2 4 に記載の圧縮方法。

【請求項 2 7】 連続する4個のサンプル点のサンプリングデータをA,B,C,Dとした場合に、注目する2個のサンプリングデータB,Cを、

$$(9(B+C) - (A+D)) / 16$$

なる演算によって1個の間引きデータに置き換え、注目位置を2サンプル点ずつずらしながら同様の演算を順次行うようにしたことを特徴とする請求項 2 4 に記載の特徴とする圧縮方法。

【請求項 2 8】 連続する7個のサンプル点のサンプリングデータをA,B,C,D,E,F,Gとした場合に、注目する1個のサンプリングデータDを、

$$(16D + 9(C+E) - (A+G)) / 32$$

なる演算によって 1 個の間引きデータに置き換え、注目位置を 2 サンプル点ずつずらしながら同様の演算を順次行うようにしたことを特徴とする請求項 2 4 に記載の特徴とする圧縮方法。

【請求項 2 9】 上記デジタル基本関数の値に応じた重み付け加算により求められた間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、上記デジタル基本関数の値に応じた重み付け加算をすることにより、上記求められた間引きデータに対する間引きデータを更に求めるようにしたことを特徴とする請求項 2 4 に記載の圧縮方法。

【請求項 3 0】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引きステップと、

上記間引きステップで求められた間引きデータを用いて、2 つの間引きデータ間を結ぶ直線上のデータ値と、その直線上のデータ値と同じサンプル点における間引きデータ値との誤差が所望の値以下となるサンプル点を標本点として検出する標本点検出ステップと、

上記標本点検出ステップで検出された各標本点における離散的な振幅データと、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得る圧縮データ生成ステップとを有することを特徴とする圧縮方法。

【請求項 3 1】 上記間引きステップの前に、上記圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換ステップを有し、

上記置換ステップで求められたデータに対して上記間引きステップの演算を行うようにしたことを特徴とする請求項 3 0 に記載の圧縮方法。

【請求項 3 2】 上記圧縮データ生成ステップで求められた上記各標本点における振幅データの下位数ビットを丸める演算を行う丸め演算ステップを有し、

上記丸め演算の施された上記各標本点における振幅データと、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得るようにしたことを

特徴とする請求項 3 0 または 3 1 に記載の圧縮方法。

【請求項 3 3】 圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換ステップと、

上記置換ステップで求められ順次出力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次出力されるサンプリングデータから間引きデータを求める間引きステップとを有することを特徴とする圧縮方法。

【請求項 3 4】 上記間引きステップで求められた間引きデータの下位数ビットを丸める演算を行う丸め演算ステップを有することを特徴とする請求項 3 3 に記載の圧縮方法。

【請求項 3 5】 上記丸め演算は、丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 3 4 に記載の圧縮方法。

【請求項 3 6】 上記丸め演算ステップで求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮ステップを有することを特徴とする請求項 3 5 に記載の圧縮方法。

【請求項 3 7】 圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引きステップと、

上記間引きステップで求められた間引きデータの下位数ビットを丸める演算を行う丸め演算ステップとを有することを特徴とする圧縮方法。

【請求項 3 8】 上記丸め演算は、丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 3 7 に記載の圧縮方法。

【請求項 3 9】 上記丸め演算ステップで求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続

数を表す値との組に置き換えて出力するゼロ圧縮ステップを有することを特徴とする請求項 3 8 に記載の圧縮方法。

【請求項 4 0】 圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換ステップと、

上記置換ステップで求められ順次出力されるサンプリングデータに対して、丸め演算前後のデータ値が非線形な関係となる演算によって下位数ビットを丸める演算を行う丸め演算ステップと、

上記丸め演算ステップで求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮ステップとを有することを特徴とする圧縮方法。

【請求項 4 1】 圧縮対象として入力されるサンプリングデータに対して、丸め演算前後のデータ値が非線形な関係となる演算によって下位数ビットを丸める演算を行う丸め演算ステップと、

上記丸め演算ステップで求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮ステップとを有することを特徴とする圧縮方法。

【請求項 4 2】 請求項 9 ～ 2 3 の何れか 1 項に記載の各手段としてコンピュータを機能させるための圧縮プログラム。

【請求項 4 3】 請求項 2 4 ～ 4 1 の何れか 1 項に記載の圧縮方法の処理手順をコンピュータに実行させるための圧縮プログラム。

【請求項 4 4】 請求項 9 ～ 2 3 の何れか 1 項に記載の各手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 4 5】 請求項 2 4 ～ 4 1 の何れか 1 項に記載の圧縮方法の処理手順をコンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 4 6】 請求項 1 に記載の圧縮装置により生成された離散的な間引きデータを入力して順次遅延させる数段の遅延回路と、

上記数段の遅延回路の各出力段から取り出されるそれぞれのデータに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間データを求める乗加算回路とを備えたことを特徴とする伸長装置。

【請求項 47】 上記数段の遅延回路と上記乗加算回路とを 1 組のオーバーサンプリング回路として、複数のオーバーサンプリング回路をカスケード接続したことを特徴とする請求項 46 に記載の伸長装置。

【請求項 48】 請求項 2 に記載の圧縮装置により生成された離散的な間引きデータを入力して順次遅延させる 3 段の遅延回路と、

上記 3 段の遅延回路の各出力段から取り出されるそれぞれのデータに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間データを求める乗加算回路とを備えたことを特徴とする伸長装置。

【請求項 49】 上記乗加算回路は、1 段目の遅延回路の出力データを  $-1$  倍する第 1 の乗算器と、

2 段目の遅延回路の出力データを 8 倍する第 2 の乗算器と、

3 段目の遅延回路の出力データを  $-1$  倍する第 3 の乗算器と、

上記 1 段目の遅延回路の出力データと上記第 1 の乗算器の出力データとを選択的に切り替えて出力する第 1 の切替回路と、

上記 3 段目の遅延回路の出力データと上記第 3 の乗算器の出力データとを選択的に切り替えて出力する第 2 の切替回路と、

上記第 2 の乗算器の出力データ、上記第 1 の切替回路の出力データおよび上記第 2 の切替回路の出力データを加算する加算器とを備えることを特徴とする請求項 48 に記載の伸長装置。

【請求項 50】 上記乗加算回路は、1 段目の遅延回路の出力データを  $-1$  倍する第 1 の乗算器と、2 段目の遅延回路の出力データを 8 倍する第 2 の乗算器と、上記第 1 の乗算器の出力データ、上記第 2 の乗算器の出力データおよび 3 段目の遅延回路の出力データを加算する加算器とから成る第 1 の乗加算回路と、

上記 2 段目の遅延回路の出力データを 8 倍する第 3 の乗算器と、上記 3 段目の遅延回路の出力データを  $-1$  倍する第 4 の乗算器と、上記第 3 の乗算器の出力データ、上記第 4 の乗算器の出力データおよび上記 1 段目の遅延回路の出力データ



を加算する加算器とから成る第2の乗加算回路と、

上記第1の乗加算回路の出力データと上記第2の乗加算回路の出力データとを選択的に切り替えて出力する切替回路とを備えることを特徴とする請求項48に記載の伸長装置。

【請求項51】 上記乗加算回路は、1段目の遅延回路の出力データを-1倍する第1の乗算器と、

2段目の遅延回路の出力データを8倍する第2の乗算器と、

3段目の遅延回路の出力データを-1倍する第3の乗算器と、

上記第1の乗算器の出力データ、上記第2の乗算器の出力データおよび上記3段目の遅延回路の出力データを加算する第1の加算器と、

上記第2の乗算器の出力データ、上記第3の乗算器の出力データおよび上記1段目の遅延回路の出力データを加算する第2の加算器と、

上記第1の加算器の出力データと上記第2の加算器の出力データとを選択的に切り替えて出力する切替回路とを備えることを特徴とする請求項48に記載の伸長装置。

【請求項52】 上記3段の遅延回路と上記乗加算回路とを1組のオーバーサンプリング回路として、複数のオーバーサンプリング回路をカスケード接続したことを特徴とする請求項48に記載の伸長装置。

【請求項53】 請求項1に記載の圧縮装置により生成された離散的な間引きデータを入力して順次遅延させる数段の遅延回路と、

上記数段の遅延回路の各出力段から取り出されるそれぞれのデータに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間データを求める乗加算回路と、

上記乗加算回路から出力される隣接する補間データどうしの平均データを求める平均化回路とを備えたことを特徴とする伸長装置。

【請求項54】 上記数段の遅延回路と上記乗加算回路と上記平均化回路とを1組のオーバーサンプリング回路として、複数のオーバーサンプリング回路をカスケード接続したことを特徴とする請求項53に記載の伸長装置。

【請求項55】 請求項2に記載の圧縮装置により生成された離散的な間引

きデータを入力して順次遅延させる 4 段の遅延回路と、

上記 4 段の遅延回路の各出力段から取り出されるそれぞれのデータに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間データを求める乗加算回路とを備えたことを特徴とする伸長装置。

【請求項 56】 上記乗加算回路は、1 段目の遅延回路の出力データを  $-1$  倍する第 1 の乗算器と、

2 段目の遅延回路の出力データを 9 倍する第 2 の乗算器と、

3 段目の遅延回路の出力データを 9 倍する第 3 の乗算器と、

4 段目の遅延回路の出力データを  $-1$  倍する第 4 の乗算器と、

上記第 1 ～ 第 4 の乗算器の出力データを加算する加算器と、

上記加算器の出力データと上記 1 段目の遅延回路に入力される間引きデータとを選択的に切り替えて出力する切替回路とを備えることを特徴とする請求項 55 に記載の伸長装置。

【請求項 57】 上記乗加算回路は、1 段目の遅延回路の出力データと 4 段目の遅延回路の出力データとを加算する第 1 の加算器と、

2 段目の遅延回路の出力データと 3 段目の遅延回路の出力データとを加算する第 2 の加算器と、

上記第 1 の加算器の出力データを  $-1$  倍する第 1 の乗算器と、

上記第 2 の加算器の出力データを 9 倍する第 2 の乗算器と、

上記第 1 の加算器の出力データと上記第 2 の加算器の出力データとを加算する第 3 の加算器と、

上記第 3 の加算器の出力データと上記 1 段目の遅延回路に入力される間引きデータとを選択的に切り替えて出力する切替回路とを備えることを特徴とする請求項 55 に記載の伸長装置。

【請求項 58】 上記 4 段の遅延回路と上記乗加算回路とを 1 組のオーバーサンプリング回路として、複数のオーバーサンプリング回路をカスケード接続したことを特徴とする請求項 55 に記載の伸長装置。

【請求項 59】 請求項 5 に記載の圧縮装置により生成された離散的な間引きデータを入力して順次遅延させる 5 段の遅延回路と、

上記 5 段の遅延回路の各出力段から取り出されるそれぞれのデータに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間データを求める乗加算回路とを備えたことを特徴とする伸長装置。

【請求項 6 0】 上記乗加算回路は、1 段目の遅延回路の出力データを  $-1$  倍する第 1 の乗算器と、2 段目の遅延回路の出力データを 9 倍する第 2 の乗算器と、3 段目の遅延回路の出力データを 25 倍する第 3 の乗算器と、4 段目の遅延回路の出力データを  $-1$  倍する第 4 の乗算器と、上記第 1 ～ 第 4 の乗算器の出力データを加算する加算器とから成る第 1 の乗加算回路と、

2 段目の遅延回路の出力データを  $-1$  倍する第 5 の乗算器と、4 段目の遅延回路の出力データを 9 倍する第 6 の乗算器と、5 段目の遅延回路の出力データを  $-1$  倍する第 7 の乗算器と、上記第 3 の乗算器および上記第 5 ～ 第 7 の乗算器の出力データを加算する加算器とから成る第 2 の乗加算回路と、

上記第 1 の乗加算回路の出力データと上記第 2 の乗加算回路の出力データとを選択的に切り替えて出力する切替回路とを備えることを特徴とする請求項 5 9 に記載の伸長装置。

【請求項 6 1】 順次入力される間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力される間引きデータに対する補間データを求めるようにしたことを特徴とする伸長装置。

【請求項 6 2】 請求項 9 ～ 1 1 の何れか 1 項に記載の圧縮装置により生成された圧縮データに関し、各標本点における振幅データとタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第 1 の補間手段と、

上記第 1 の補間手段により求められた第 1 の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第 1 の補間データに対する第 2 の補間データを更に求める第 2 の補間手段とを備えることを特徴とする伸長装置。

【請求項 6 3】 請求項 1 2 ～ 1 4 の何れか 1 項に記載の圧縮装置により生

成された圧縮データ中に含まれる各標本点の振幅データに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算手段と、

上記逆丸め演算手段より出力された振幅データと上記圧縮データ中に含まれるタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第1の補間手段と、

上記第1の補間手段により求められた第1の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第1の補間データに対する第2の補間データを更に求める第2の補間手段とを備えることを特徴とする伸長装置。

【請求項64】 上記丸め演算と逆の演算は、逆丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項63に記載の伸長装置。

【請求項65】 上記非線形な関係となる演算は、指数関数もしくはそれに近似した関数に基づく演算であることを特徴とする請求項64に記載の伸長装置。

【請求項66】 請求項15に記載の圧縮装置により生成された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間手段を備えることを特徴とする伸長装置。

【請求項67】 請求項16、17、19または20に記載の圧縮装置により生成された丸め演算後の離散的な間引きデータに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算手段と、

上記逆丸め演算手段より出力された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間手段とを備えることを特徴とする伸長装置。

【請求項 68】 請求項 18 または 21 に記載の圧縮装置により生成された圧縮データについて、 $-0$  の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長手段と、

上記ゼロ伸長手段により上記ゼロデータの復元が行われた離散的な間引きデータに対して、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算手段と、

上記逆丸め演算手段より出力された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間手段とを備えることを特徴とする伸長装置。

【請求項 69】 請求項 22 または 23 に記載の圧縮装置により生成された圧縮データについて、 $-0$  の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長手段と、

上記ゼロ伸長手段により上記ゼロデータの復元が行われた離散的な間引きデータに対して、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算手段とを備えることを特徴とする伸長装置。

【請求項 70】 請求項 24 に記載の圧縮方法により生成された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求めるようにしたことを特徴とする伸長方法。

【請求項 71】 上記離散的な間引きデータを、上記デジタル基本関数の値に応じた重み付け加算により求めた 2 つの補間データにそれぞれ置き換えるようにしたことを特徴とする請求項 70 に記載の伸長方法。

【請求項 72】 上記デジタル基本関数の値に応じた重み付け加算により求められた補間データについて、隣接する補間データどうしで平均化演算を行うようにしたことを特徴とする請求項 71 に記載の伸長方法。

【請求項 73】 上記デジタル基本関数の重み付け加算により求められた補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル

点の補間データとに対して、上記デジタル基本関数の値に応じた重み付け加算をすることにより、上記注目するサンプル点の補間データに対する補間データを更に求めるようにしたことを特徴とする請求項 7 0 に記載の伸長方法。

【請求項 7 4】 請求項 3 0 または 3 1 に記載の圧縮方法により生成された圧縮データに関し、各標本点における振幅データとタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第 1 の補間ステップと、

上記第 1 の補間ステップで求められた第 1 の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第 1 の補間データに対する第 2 の補間データを更に求める第 2 の補間ステップとを有することを特徴とする伸長方法。

【請求項 7 5】 請求項 3 2 に記載の圧縮方法により生成された圧縮データ中に含まれる各標本点の振幅データに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算ステップと、

上記逆丸め演算ステップで求められ順次出力される振幅データと上記圧縮データ中に含まれるタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第 1 の補間ステップと、

上記第 1 の補間ステップで求められた第 1 の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第 1 の補間データに対する第 2 の補間データを更に求める第 2 の補間ステップとを有することを特徴とする伸長方法。

【請求項 7 6】 上記丸め演算と逆の演算は、逆丸め演算前後のデータ値が非線形な関係となる演算によって行うことを特徴とする請求項 7 5 に記載の伸長方法。

【請求項 7 7】 請求項 3 3 に記載の圧縮方法により生成された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算

をすることにより、上記離散的な間引きデータに対する補間データを求めるようにしたことを特徴とする伸長方法。

【請求項 78】 請求項 34、35、37 または 38 に記載の圧縮方法により生成された丸め演算後の離散的な間引きデータに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算ステップと、

上記逆丸め演算ステップで求められた逆丸め演算後の離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間ステップとを有することを特徴とする伸長方法。

【請求項 79】 請求項 36 または 39 に記載の圧縮方法により生成された圧縮データについて、-0 の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長ステップと、

上記ゼロ伸長ステップで上記ゼロデータの復元が行われた離散的な間引きデータに対して、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算ステップと、

上記逆丸め演算ステップで求められ順次出力された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間ステップとを有することを特徴とする伸長方法。

【請求項 80】 請求項 40 または 41 に記載の圧縮方法により生成された圧縮データについて、-0 の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長ステップと、

上記ゼロ伸長ステップで上記ゼロデータの復元が行われた離散的な間引きデータに対して、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算ステップとを有することを特徴とする伸長方法。

【請求項 81】 請求項 62～69 の何れか 1 項に記載の各手段としてコンピュータを機能させるための伸長プログラム。

【請求項 8 2】 請求項 7 0 ～ 8 0 の何れか 1 項に記載の伸長方法の処理手順をコンピュータに実行させるための伸長プログラム。

【請求項 8 3】 請求項 6 2 ～ 6 9 の何れか 1 項に記載の各手段としてコンピュータを機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 8 4】 請求項 7 0 ～ 8 0 の何れか 1 項に記載の伸長方法の処理手順をコンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 8 5】 圧縮側において、圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるとともに、

伸長側において、順次入力される間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、上記デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力される間引きデータに対する補間データを求めるようにしたことを特徴とする圧縮伸長システム。

【請求項 8 6】 上記圧縮側において、上記圧縮対象として順次入力されるサンプリングデータの中で絶対値が所定値より小さいデータをゼロデータに置き換える処理を行い、置き換え後のデータに対して上記デジタル基本関数の値に応じた重み付け加算を行うことによって上記間引きデータを求めるようにしたことを特徴とする請求項 8 5 に記載の圧縮伸長システム。

【請求項 8 7】 上記圧縮側において、上記求められた間引きデータの下位数ビットを丸める丸め演算を行い、

上記伸長側において、上記順次入力される間引きデータに対して上記丸め演算と逆の演算を行い、逆丸め演算後のデータに対して上記デジタル基本関数の値に応じた重み付け加算を行うことによって上記補間データを求めるようにしたことを特徴とする請求項 8 5 または 8 6 に記載の圧縮伸長システム。



【請求項 8 8】 上記圧縮側における丸め演算は、逆丸め演算前後のデータ値が非線形な関係となる演算であり、上記求められた丸め演算後の間引きデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを - 0 の値とゼロデータの連続数を表す値との組に置き換えて出力するようにしたことを特徴とする請求項 8 7 に記載の圧縮伸長システム。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は圧縮装置及び方法、伸長装置及び方法、圧縮伸長システム、プログラム、記録媒体に関し、特に、音声や画像など、連続的なアナログ信号もしくは離散的なデジタル信号を圧縮および伸長する方式に関するものである。

【 0 0 0 2 】

【従来の技術】

従来、画像信号や音声信号など情報量の多い信号を伝送したり蓄積したりする場合に、伝送情報量の削減や、蓄積メディアへの保存可能時間の長時間化等を目的として、信号を圧縮・伸長することが行われている。一般に、アナログ信号を圧縮する場合、まず所定のサンプリング周波数に従ってアナログ信号をサンプリングしてデジタル化し、得られたデジタルデータに対して圧縮処理を行う。

【 0 0 0 3 】

圧縮処理の一例として、間引き圧縮が存在する。間引き圧縮の最も簡単な方法は、全てのサンプル点のうち所定のサンプル点のデータを破棄する（逆に言えば、所定のサンプル点のデータのみを採用する）ようにしたものである。例えば、各サンプル点のデータを 1 個飛びに採用していくことで、2 倍の圧縮率が達成される。圧縮率を高めたい場合は、各サンプル点のデータを  $n$  ( $> 2$ ) 個飛びに採用すれば良い。

【 0 0 0 4 】

この方法によれば、演算を全く行うことなく信号を圧縮することが可能であるが、伸長による元データへの再現性が問題となる。すなわち、伸長側では、圧縮時に採用されたサンプル点間のデータを補間演算により求めることになるが、採

用するサンプル点の間隔を広くして圧縮率を高めるほど、元データを正確に再現するためには本来必要であったサンプル点のデータが多く破棄されてしまうため、元データへの再現性は著しく悪化する。

#### 【 0 0 0 5 】

そのため、圧縮率の向上と元データへの再現性との両立を目的として、各サンプル点のデータに何らかの演算を行うことによって情報量を間引くのが通常である。例えば、隣接する 2 つのサンプル点のデータ A, B を、その平均データ  $(A + B) / 2$  に置き換える方法が考えられる。このようにすれば、元データ A, B の何れか一方を完全に破棄してしまう場合と異なり、元データ A, B の両方が加味されたデータが圧縮データ（間引きデータ）として採用されるため、元データへの再現性は多少向上する。

#### 【 0 0 0 6 】

しかし、この程度の演算では、実用に耐え得る良好な再現性は得ることができない。圧縮率を高めるとともに実用に耐え得るデータ再現性を得るためには、より複雑な演算が必要になる。

また、間引き圧縮を行うと、周波数特性の劣化からいわゆる折り返しノイズが発生してしまい、伸長によって得られる信号の品質を劣化させる要因となる。

#### 【 0 0 0 7 】

このように、単純な間引き圧縮によって圧縮率の向上と再生データの品質向上との両方を実現することは、極めて困難であった。このような実情から、現在では、単なる間引き圧縮ではない複雑な演算を行う圧縮・伸長の方式が多く用いられている。

#### 【 0 0 0 8 】

例えば、画像信号や音声信号の圧縮においては、D C T (Discreat-Cosine-Transform) 等の時間軸－周波数軸の変換フィルタを用いて元のデータを加工した後に、周波数領域で圧縮を行う手法が用いられている。音声信号の圧縮方式として電話回線で良く用いられる D P C M (Differential Pulse Code Modulation) も、この点を意図して使用している。なお、この D P C M による圧縮方式は、波形をサンプリングするとき隣り合うサンプル値の差分を符号化する方式である。

**【 0 0 0 9 】**

また、時間／周波数変換を行う方式としては、サブバンドフィルタやM D C T (Modified Discrete Cosine Transform)を用いた方式もあり、このような方式を用いた符号化方式としてM P E G (Moving Picture Image Coding Experts Group) オーディオが挙げられる。

また、最も広く使用されている画像の圧縮システムも、このM P E G規格として一般的に知られている。

**【 0 0 1 0 】**

上述の圧縮方式に従って圧縮されたデータの伸長処理は、基本的には同じ圧縮方式の圧縮処理と逆の操作によって行われる。

すなわち、圧縮されたデジタルデータは、周波数／時間変換処理によって周波数領域の信号から時間領域の信号に変換された後、所定の伸長処理が施されることにより、元のデジタルデータが再現される。そして、このようにして求められた元データが、必要に応じてデジタル－アナログ変換され、アナログ信号として出力される。

**【 0 0 1 1 】****【発明が解決しようとする課題】**

近年において、画像信号や音声信号の高精細化等に伴う情報量の増加、移動端末やインターネットを利用した情報通信の普及が進み、高圧縮率化および再生データの高品質化への要求がますます高まっている。しかしながら、D C T等を利用した上記従来の圧縮・伸長方式では、改良が加えられてきているものの、更に高い圧縮率と高品質な再生データを得るには限界があり、更なる高圧縮率化と高品質化を実現することは極めて困難であった。

**【 0 0 1 2 】**

また、上記従来の圧縮・伸長方式では、時間軸上の信号を周波数軸上の信号に変換して圧縮するようにしているので、圧縮の際の時間／周波数変換および伸長の際の周波数／時間変換などの処理が必要となる。そのため、処理が煩雑化するとともに、これを実現するための構成が非常に複雑になるという問題があった。これは、圧縮・伸長にかかる処理時間が長くなるだけでなく、装置の小型化を困

難にする要因となっていた。

### 【0 0 1 3】

本発明は、このような問題を解決するために成されたものであり、圧縮率の向上と再生データの品質向上との両方を実現する全く新しい圧縮・伸長方式を提供することを目的とする。

また、本発明は、信号の圧縮・伸長処理を簡素化して処理時間を短くすることができるようになるとともに、これを実現するための構成も簡単化できるようにすることをも目的としている。

### 【0 0 1 4】

#### 【課題を解決するための手段】

本発明の圧縮装置は、順次入力されるサンプリングデータを順次遅延させる数段の遅延回路と、上記数段の遅延回路の各出力段から取り出されるそれぞれのデータに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める乗加算回路とを備えたことを特徴とする。

### 【0 0 1 5】

本発明の他の態様では、上記数段の遅延回路と上記乗加算回路とを1組の間引き回路として、複数の間引き回路をカスケード接続したことを特徴とする。

### 【0 0 1 6】

本発明のその他の態様では、圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるようにしたことを特徴とする。

### 【0 0 1 7】

本発明のその他の態様では、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引き手段と、上記間引き手段により求められた間

引きデータを用いて、2つの間引きデータ間を結ぶ直線上のデータ値と、その直線上のデータ値と同じサンプル点における間引きデータ値との誤差が所望の値以下となるサンプル点を標本点として検出する標本点検出手段と、上記標本点検出手段により検出された各標本点における離散的な振幅データと、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得る圧縮データ生成手段とを備えることを特徴とする。

#### 【0018】

本発明のその他の態様では、圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換手段を備えたことを特徴とする。

#### 【0019】

本発明のその他の態様では、各標本点あるいは各サンプル点における振幅データの下位数ビットを丸める演算を行う丸め演算手段を備えたことを特徴とする。

ここで、上記丸め演算は、例えば、丸め演算前後のデータ値が非線形な関係となる演算によって行う。

#### 【0020】

本発明のその他の態様では、上記丸め演算手段により求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを-0の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮手段を備えることを特徴とする。

#### 【0021】

また、本発明の圧縮方法は、圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるようにしたことを特徴とする。

#### 【0022】

本発明の他の態様では、圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点

のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求める間引きステップと、上記間引きステップで求められた間引きデータを用いて、2つの間引きデータ間を結ぶ直線上のデータ値と、その直線上のデータ値と同じサンプル点における間引きデータ値との誤差が所望の値以下となるサンプル点を標本点として検出する標本点検出ステップと、上記標本点検出ステップで検出された各標本点における離散的な振幅データと、各標本点間の時間間隔を表すタイミングデータとの組を圧縮データとして得る圧縮データ生成ステップとを有することを特徴とする。

#### 【 0 0 2 3 】

本発明のその他の態様では、圧縮対象として入力されるサンプリングデータ中で絶対値が所定値より小さいサンプリングデータをゼロデータに置き換える処理を行う置換ステップを有することを特徴とする。

#### 【 0 0 2 4 】

本発明のその他の態様では、各標本点あるいは各サンプル点における振幅データの下位数ビットを丸める演算を行う丸め演算ステップを有することを特徴とする。

ここで、上記丸め演算は、例えば、丸め演算前後のデータ値が非線形な関係となる演算によって行う。

#### 【 0 0 2 5 】

本発明のその他の態様では、上記丸め演算ステップで求められたデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを-0の値とゼロデータの連続数を表す値との組に置き換えて出力するゼロ圧縮ステップを有することを特徴とする。

#### 【 0 0 2 6 】

また、本発明の伸長装置は、請求項1に記載の圧縮装置により生成された離散的な間引きデータを入力して順次遅延させる数段の遅延回路と、上記数段の遅延回路の各出力段から取り出されるそれぞれのデータに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記間引きデータに対する補間デー

タを求める乗加算回路とを備えたことを特徴とする。

【0027】

本発明の他の態様では、上記数段の遅延回路と上記乗加算回路とを1組のオーバーサンプリング回路として、複数のオーバーサンプリング回路をカスケード接続したことを特徴とする。

【0028】

本発明のその他の態様では、上記乗加算回路から出力される隣接する補間データどうしの平均データを求める平均化回路とを備えたことを特徴とする。

【0029】

本発明のその他の態様では、順次入力される間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対し、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力される間引きデータに対する補間データを求めるようにしたことを特徴とする。

【0030】

本発明のその他の態様では、請求項9～11の何れか1項に記載の圧縮装置により生成された圧縮データに関し、各標本点における振幅データとタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第1の補間手段と、上記第1の補間手段により求められた第1の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第1の補間データに対する第2の補間データを更に求める第2の補間手段とを備えることを特徴とする。

【0031】

本発明のその他の態様では、圧縮データ中に含まれる各標本点あるいは各サンプル点の振幅データに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算手段を備えることを特徴とする。

【0032】

本発明のその他の態様では、請求項15に記載の圧縮装置により生成された離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数

個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求める補間手段を備えることを特徴とする。

#### 【 0 0 3 3 】

本発明のその他の態様では、 $-0$ の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長手段を備えることを特徴とする。

#### 【 0 0 3 4 】

また、本発明の伸長方法は、離散的な間引きデータについて、注目するサンプル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求めるようにしたことを特徴とする。

#### 【 0 0 3 5 】

本発明の他の態様では、上記デジタル基本関数の値に応じた重み付け加算により求められた補間データについて、隣接する補間データどうしで平均化演算を行うようにしたことを特徴とする。

#### 【 0 0 3 6 】

本発明のその他の態様では、各標本点における振幅データとタイミングデータとを用いて、上記タイミングデータによって示される時間間隔を有する振幅データの間を補間する第1の補間ステップと、上記第1の補間ステップで求められた第1の補間データについて、注目するサンプル点の補間データとその周辺数個のサンプル点の補間データとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記第1の補間データに対する第2の補間データを更に求める第2の補間ステップとを有することを特徴とする。

#### 【 0 0 3 7 】

本発明のその他の態様では、圧縮データ中に含まれる各標本点の振幅データに対し、圧縮時に行われた丸め演算と逆の演算を行う逆丸め演算ステップを有することを特徴とする。

#### 【 0 0 3 8 】

本発明のその他の態様では、離散的な間引きデータについて、注目するサン



ル点の間引きデータとその周辺数個のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記離散的な間引きデータに対する補間データを求めるようにしたことを特徴とする。

#### 【 0 0 3 9 】

本発明のその他の態様では、 $-0$ の値を検知したときに、それに対応する連続数の分だけゼロデータを復元するゼロ伸長ステップを有することを特徴とする。

#### 【 0 0 4 0 】

また、本発明のプログラムは、請求項 9 ～ 2 3 の何れか 1 項に記載の各手段としてコンピュータを機能させるための圧縮プログラム、請求項 2 4 ～ 4 1 の何れか 1 項に記載の圧縮方法の処理手順をコンピュータに実行させるための圧縮プログラム、請求項 6 2 ～ 6 9 の何れか 1 項に記載の各手段としてコンピュータを機能させるための伸長プログラム、あるいは、請求項 7 0 ～ 8 0 の何れか 1 項に記載の伸長方法の処理手順をコンピュータに実行させるための伸長プログラムである。

#### 【 0 0 4 1 】

また、本発明のコンピュータ読み取り可能な記録媒体は、請求項 9 ～ 2 3 の何れか 1 項に記載の各手段としてコンピュータを機能させるためのプログラム、請求項 2 4 ～ 4 1 の何れか 1 項に記載の圧縮方法の処理手順をコンピュータに実行させるためのプログラム、請求項 6 2 ～ 6 9 の何れか 1 項に記載の各手段としてコンピュータを機能させるためのプログラム、あるいは、請求項 7 0 ～ 8 0 の何れか 1 項に記載の伸長方法の処理手順をコンピュータに実行させるためのプログラムを記録したことを特徴とする。

#### 【 0 0 4 2 】

また、本発明の圧縮伸長システムは、圧縮側において、圧縮対象として順次入力されるサンプリングデータについて、注目するサンプル点のサンプリングデータとその周辺数個のサンプル点のサンプリングデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力されるサンプリングデータから間引きデータを求めるとともに、伸長側において、順次入力される間引きデータについて、注目するサンプル点の間引きデータとその周辺数個の

サンプル点の間引きデータとに対して、上記デジタル基本関数の値に応じた重み付け加算をすることにより、上記順次入力される間引きデータに対する補間データを求めるようにしたことを特徴とする。

#### 【0043】

本発明のその他の態様では、上記圧縮側において、上記圧縮対象として順次入力されるサンプリングデータの中で絶対値が所定値より小さいデータをゼロデータに置き換える処理を行い、置き換え後のデータに対して上記デジタル基本関数の値に応じた重み付け加算を行うことによって上記間引きデータを求めるようにしたことを特徴とする。

#### 【0044】

本発明のその他の態様では、上記圧縮側において、上記求められた間引きデータの下位数ビットを丸める丸め演算を行い、上記伸長側において、上記順次入力される間引きデータに対して上記丸め演算と逆の演算を行い、逆丸め演算後のデータに対して上記デジタル基本関数の値に応じた重み付け加算を行うことによって上記補間データを求めるようにしたことを特徴とする。

#### 【0045】

本発明のその他の態様では、上記圧縮側における丸め演算は、逆丸め演算前後のデータ値が非線形な関係となる演算であり、上記求められた丸め演算後の間引きデータについて、絶対値ゼロのデータが所定数以上続いたときに、それを－0の値とゼロデータの連続数を表す値との組に置き換えて出力するようにしたことを特徴とする。

#### 【0046】

##### 【発明の実施の形態】

##### （第1の実施形態）

以下、本発明の第1の実施形態を図面に基づいて説明する。

図1は、第1の実施形態による圧縮装置（間引き装置）の構成例を示す図である。図2は、第1の実施形態による伸長装置（補間装置）の構成例を示す図である。図3は、第1の実施形態による間引きおよび補間の原理を説明するための図であり、（a）は間引きの原理を示し、（b）は補間の原理を示している。

**【 0 0 4 7 】**

まず、図 3 (a) を用いて本実施形態による間引きの原理について説明する。図 3 (a) に示すデータ構造において、横軸の A, B, C, … は 1 クロック毎に順次入力される各サンプル点のサンプリングデータを示し、そのデータ値がデジタル基本関数に対する係数となる。また、縦軸の a, b, c, … は、デジタル基本関数により加工されたサンプリングデータの中心位置を示す。

**【 0 0 4 8 】**

ここで用いるデジタル基本関数は、オーバーサンプリングによるデータ補間を行う際に使用する標本化関数の基本となるものであり、図 4 に示されるものである。このデジタル基本関数は、1 クロック毎にデータ値を  $-1, 1, 8, 8, 1, -1$  と変化させて作成したものである。

**【 0 0 4 9 】**

図 3 に示すように、サンプリングデータ A に対するデジタル基本関数は、最初のクロック位置を先頭として各関数値 ( $-1, 1, 8, 8, 1, -1$ ) を 1 クロックずつずらして与えられる。また、次のサンプリングデータ B に対するデジタル基本関数は、最初のクロック位置より 2 クロック後のクロック位置を先頭として各関数値を 1 クロックずつずらして与えられる。同様に、順次入力される各サンプリングデータ C, D, E, F, … に対するデジタル基本関数は、更に 2 クロックずつ遅れたクロック位置をそれぞれ先頭として、各関数値を 1 クロックずつずらして与えられる。

**【 0 0 5 0 】**

本実施形態では、このようなデータ構造に基づいて、各サンプル点のサンプリングデータ A, B, C, … を  $1/2$  に間引く演算を行う。すなわち、2 つの隣接するサンプリングデータ (B, C), (D, E), (F, G), … どうしで所定の演算をしてそれぞれ 1 つの間引きデータ  $(B+C)/2, (D+E)/2, (F+G)/2, …$  に置き換える処理を行う。

**【 0 0 5 1 】**

ここでは、2 つの隣接するサンプリングデータ (B, C) から 1 つの間引きデータ ( $(B+C)/2$ ) を生成する場合を例に挙げて説明する。デジタル基本関数により加工さ

れたサンプリングデータBの中心位置bを挟む2行のデータをそれぞれb1, b2とすると、

$$b1 = A + 8B - C \quad \dots(1)$$

$$b2 = -A + 8B + C \quad \dots(2)$$

と表される。

### 【0 0 5 2】

上記式(1)(2)より、

$$b = (b1 + b2) / 2 = 8B、つまり、B = b / 8 \quad \dots(3)$$

が成り立つ。同様に、サンプリングデータCの中心位置cのデータに関して、

$$c1 = B + 8C - D \quad \dots(4)$$

$$c2 = -B + 8C + D \quad \dots(5)$$

$$c = (c1 + c2) / 2 = 8C、つまり、C = c / 8 \quad \dots(6)$$

が成り立つ。

### 【0 0 5 3】

一方、2つの隣接するサンプリングデータ(B, C)に対する1つの間引きデータ  $(B + C) / 2$  は、データb2とデータc1との畳み込み演算により求める。すなわち、上記式(2)(3)(4)(6)より、間引きデータは、

$$\begin{aligned} (B + C) / 2 &= (b2 + c1) / 16 \\ &= ((-A + 8B + C) + (B + 8C - D)) / 16 \\ &= (9(B + C) - (A + D)) / 16 \quad \dots(7) \end{aligned}$$

と表される。

### 【0 0 5 4】

このように、2つのサンプリングデータB, Cは、式(7)で示される1つの間引きデータ  $(B + C) / 2$  に置き換えることができる。この間引きデータは、注目している2つのサンプリングデータB, Cを加算して9 / 16倍した値から、その両側に隣接する2つのサンプリングデータA, Dを加算して1 / 16倍した値を引いたものである。

### 【0 0 5 5】

これと同様にして、2つのサンプリングデータ(D, E)を1つの間引きデータ(D

+E)/2に置き換える。以降のサンプリングデータ(F,G),…についても同様である。なお、最初のサンプリングデータAについては、同様の演算を行うために必要なデータが不足していることから、間引きを行うことなくそのままの状態とする。

#### 【0056】

このようにして1/2に間引いたデータをもとに同様の間引き演算を行うことにより、元データを1/4に間引くことが可能である。さらに、1/4に間引いたデータをもとに同様の間引き演算を行うことにより、元データを1/8に間引くことが可能である。理論的には、同様の間引き演算をn回繰り返し行うことにより、元データを1/2<sup>n</sup>倍に圧縮することが可能である。

#### 【0057】

次に、図3(b)を用いて本実施形態によるデータ補間の原理について説明する。図3(b)に示すデータ構造において、横軸のA,B,C,…は1クロック毎に順次入力される間引きデータを示し、そのデータ値がデジタル基本関数に対する係数となる。また、縦軸のa,b,c,…は、デジタル基本関数により加工された間引きデータの中心位置を示す。ここで用いるデジタル基本関数も、図4に示されるものである。

#### 【0058】

図3(b)に示すように、間引きデータAに対するデジタル基本関数は、最初のクロック位置を先頭として各関数値(-1, 1, 8, 8, 1, -1)を1クロックずつずらして与えられる。次の間引きデータBに対するデジタル基本関数は、最初のクロック位置より2クロック後のクロック位置を先頭として各関数値を1クロックずつずらして与えられる。同様に、順次入力される各間引きデータC, D,E,F, …に対するデジタル基本関数は、更に2クロックずつ遅れたクロック位置をそれぞれ先頭として、各関数値を1クロックずつずらして与えられる。

#### 【0059】

このようなデータ構造に基づいて、1つの間引きデータBから2つの補間データB1,B2を生成する場合を例に挙げて説明する。ここでは、デジタル基本関数により加工された間引きデータBの中心位置bを挟む2行のデータをそれぞれb1,b2

とし、これらのデータ  $b_1, b_2$  に対して畳み込み演算を行うことを考える。この場合、2つの隣接するデータ値  $b_1, b_2$  は、それぞれ

$$b_1 = A + 8B - C \quad \cdots (8)$$

$$b_2 = -A + 8B + C \quad \cdots (9)$$

と表される。

### 【 0 0 6 0 】

上記式 (8) (9) より、

$$b_1 + b_2 = 16B \quad \cdots (10)$$

と表され、さらに、

$$B = (b_1 / 8 + b_2 / 8) / 2 \quad \cdots (11)$$

のように変形される。

### 【 0 0 6 1 】

上記式 (11) から、間引きデータ  $B$  は、2つのデータ  $b_1 / 8, b_2 / 8$  の中間値に相当すると言える。これを逆に考えることで、1つの間引きデータ  $B$  は、2つの補間データ  $B_1, B_2$  に置き換えることができ、それらは、

$$B_1 = (A + 8B - C) / 8 \quad \cdots (12)$$

$$B_2 = (-A + 8B + C) / 8 \quad \cdots (13)$$

のように表される。

### 【 0 0 6 2 】

また、間引きデータ  $B$  について式 (11) に示される関係が他の間引きデータ  $C, D, E, F, \cdots$  についても同様に成り立つことから、

$$C = (c_1 / 8 + c_2 / 8) / 2$$

$$D = (d_1 / 8 + d_2 / 8) / 2$$

$$E = (e_1 / 8 + e_2 / 8) / 2$$

$$F = (f_1 / 8 + f_2 / 8) / 2$$

.....

となる。

### 【 0 0 6 3 】

したがって、次式に示すように、1つの間引きデータ  $C$  は2つの補間データ  $C_1$

、C2に、1つの間引きデータDは2つの補間データD1、D2に、1つの間引きデータEは2つの補間データE1、E2に、1つの間引きデータFは2つの補間データF1、F2にそれぞれ置き換えられる。

$$C \rightarrow C1 = (B + 8C - D) / 8, C2 = (-B + 8C + D) / 8$$

$$D \rightarrow D1 = (C + 8D - E) / 8, D2 = (-C + 8D + E) / 8$$

$$E \rightarrow E1 = (D + 8E - F) / 8, E2 = (-D + 8E + F) / 8$$

$$F \rightarrow F1 = (E + 8F - G) / 8, F2 = (-E + 8F + G) / 8$$

.....

#### 【0 0 6 4】

以上のように、あるサンプル点の間引きデータ（例えばB）に関して補間値を求める際には、その注目するサンプル点の間引きデータを8倍した値と、前後のサンプル点の間引きデータを+1倍、-1倍した値とを加算し、その加算値を8で割ることによって第1の補間値（例えばB1）を得る。また、注目する間引きデータを8倍した値と、前後のサンプル点の間引きデータを-1倍、+1倍した値とを加算し、その加算値を8で割ることによって第2の補間値（例えばB2）を得る。これを各サンプル点の間引きデータについて行うことにより、元の間引きデータを2倍にオーバーサンプルする。

#### 【0 0 6 5】

また、2倍にオーバーサンプルされた補間データをもとに同様の補間処理を行うことにより、元の間引きデータを4倍にオーバーサンプルすることが可能である。さらに、4倍にオーバーサンプルされた補間データをもとに同様の補間処理を行うことにより、元の間引きデータを8倍にオーバーサンプルすることが可能である。理論的には、同様の補間処理をn回繰り返すことにより、元の間引きデータを $2^n$ 倍にオーバーサンプルすることが可能である。

#### 【0 0 6 6】

圧縮側において間引き演算をn回繰り返す行い、元データを $1/2^n$ 倍に圧縮した場合は、伸長側において補間演算をn回繰り返すことにより、入力された間引きデータを $2^n$ 倍にオーバーサンプルして、圧縮前の元データを再現することが可能である。

**【0 0 6 7】**

次に、図 1 を用いて、上述の間引き動作を実現する第 1 の実施形態による圧縮装置（間引き装置）の構成について説明する。

図 1 に示す圧縮装置は、1 / 2 倍の間引きを行うための回路を 3 段カスケード接続した構成になっており、1 段目～3 段目の回路構成を符号の添数字（-1～-3）により区別している。また、添数字のみが異なり主符号が同じ回路構成は互いに同一の機能を有するものであることを示している。

**【0 0 6 8】**

1 段目の間引き回路は、基準周波数（例えば 4 4 . 1 K H z）のクロック 8 C K に従って動作する。4 段の D 型フリップフロップ 1 \_1, 2 \_1, 3 \_1, 4 \_1 は、離散的に順次入力されるサンプリングデータ（例えば 1 6 ビット）を基準周波数の 1 クロック 8 C K ずつ順に遅延させる。これらの D 型フリップフロップ 1 \_1～4 \_1 は、請求項 2 に記載した 4 段の遅延回路に相当する。

**【0 0 6 9】**

加算器 5 \_1 は、2 段目と 3 段目の D 型フリップフロップ 2 \_1, 3 \_1 の出力タップから取り出されたデータを加算する。乗算器 6 \_1 は、上記加算器 5 \_1 の出力データを 9 倍する。上記加算器 5 \_1 および乗算器 6 \_1 は、請求項 3 に記載した第 1 の乗加算回路に相当する。

**【0 0 7 0】**

加算器 7 \_1 は、1 段目と 4 段目の D 型フリップフロップ 1 \_1, 4 \_1 の出力タップから取り出されたデータを加算する。乗算器 8 \_1 は、上記加算器 7 \_1 の出力データを - 1 倍する。上記加算器 7 \_1 および乗算器 8 \_1 は、請求項 3 に記載した第 2 の乗加算回路に相当する。

**【0 0 7 1】**

加算器 9 \_1 は、上記 2 つの乗算器 6 \_1, 8 \_1 の出力データを加算する。乗算器 1 0 \_1 は、上記加算器 9 \_1 の出力データを 1 / 1 6 倍する。上記加算器 9 \_1 および乗算器 1 0 \_1 は、請求項 3 に記載した第 3 の乗加算回路に相当する。

**【0 0 7 2】**

以上のような回路に対して、図 3（a）に示すようなサンプリングデータ A, B,



C, …を順に入力することにより、乗算器 10<sub>-1</sub>からは、式(7)のように示される  $1/2$  間引きデータが出力される。乗算器 10<sub>-1</sub>の出力データは、2 段目の間引き回路の D 型フリップフロップ 1<sub>-2</sub>に入力され、2 段目の  $1/2$  間引きが実行される。2 段目の間引き回路は、基準周波数の  $1/2$  倍の周波数のクロック 4CK に従って動作すること以外は、1 段目の間引き回路と同様である。

#### 【0073】

なお、図 1 の回路によれば、式(7)に示す間引きデータ  $(B+C)/2$  が乗算器 10<sub>-1</sub> から出力された後は、データ  $(C+D)/2$  が出力される。しかし、このデータ  $(C+D)/2$  は、2 段目の間引き回路が基準周波数の  $1/2$  倍の周波数のクロック 4CK に従って動作しているために無視され、次の出力データ  $(D+E)/2$  が間引きデータとして引き続き 2 段目の間引き回路で処理される。

#### 【0074】

2 段目の間引き回路の最終段に設けられた乗算器 10<sub>-2</sub>の出力データは、3 段目の間引き回路の D 型フリップフロップ 1<sub>-3</sub>に入力され、3 段目の  $1/2$  間引きが実行される。3 段目の間引き回路は、基準周波数の  $1/4$  倍の周波数のクロック 2CK に従って動作すること以外は、1 段目の間引き回路と同様である。3 段目の間引き回路より出力されたデータは、基準周波数の  $1/8$  倍の周波数のクロック CK に従って D 型フリップフロップ 11 に保持された後、最終的な間引きデータ（圧縮データ）として出力される。

#### 【0075】

次に、図 2 を用いて、上述の補間動作を実現する第 1 の実施形態による伸長装置（補間装置）の構成について説明する。

図 2 に示す伸長装置は、2 倍のオーバーサンプリングを行うための回路を 3 段カスケード接続した構成になっており、1 段目～3 段目の回路構成を符号の添数字（-1～-3）により区別している。また、添数字のみが異なり主符号が同じ回路構成は互いに同一の機能を有するものであることを示している。

#### 【0076】

1 段目のオーバーサンプリング回路は、基準周波数の  $1/8$  倍の周波数（5.5125 KHz）のクロック CK に従って動作する。3 段の D 型フリップフロップ

プ 2 1<sub>-1</sub>, 2 2<sub>-1</sub>, 2 3<sub>-1</sub>は、離散的に順次入力される間引きデータを 1 クロック C K ずつ順に遅延させる。これらの D 型フリップフロップ 2 1<sub>-1</sub>, 2 2<sub>-1</sub>, 2 3<sub>-1</sub>は、請求項 4 8 に記載した 3 段の遅延回路に相当する。

#### 【 0 0 7 7 】

1 段目の D 型フリップフロップ 2 1<sub>-1</sub>の出力タップから取り出されるデータは、例えば式(12)(13)に示す第 1 項を形成し、2 段目の D 型フリップフロップ 2 2<sub>-1</sub>の出力タップから取り出されるデータは第 2 項を形成し、3 段目の D 型フリップフロップ 2 3<sub>-1</sub>の出力タップから取り出されるデータは第 3 項を形成する。

#### 【 0 0 7 8 】

1 段目の D 型フリップフロップ 2 1<sub>-1</sub>の出力データは、- 1 倍の乗算器 2 4<sub>-1</sub>（請求項 4 9 の第 1 の乗算器に相当）を介して AND ゲート 2 5<sub>-1</sub>の一方の入力端に入力されるとともに、乗算器を介することなく（+ 1 倍に相当）AND ゲート 2 6<sub>-1</sub>の一方の入力端に入力される。AND ゲート 2 5<sub>-1</sub>の他方の入力端には、インバータ 2 7<sub>-1</sub>を通過した反転クロック C K が入力される。また、AND ゲート 2 6<sub>-1</sub>の他方の入力端には、クロック C K が入力される。

#### 【 0 0 7 9 】

上記 2 つの AND ゲート 2 5<sub>-1</sub>, 2 6<sub>-1</sub>の出力データは、OR ゲート 2 8<sub>-1</sub>を介して出力される。これにより、クロック C K が“H”の期間中は + 1 倍された間引きデータが OR ゲート 2 8<sub>-1</sub>から出力される。また、クロック C K が“L”の期間中は - 1 倍された間引きデータが OR ゲート 2 8<sub>-1</sub>から出力される。すなわち、クロック C K が“H”の期間中に上記式(12)に示す第 1 項が求められ、“L”の期間中に上記式(13)に示す第 1 項が求められる。2 つの AND ゲート 2 5<sub>-1</sub>, 2 6<sub>-1</sub>, インバータ 2 7<sub>-1</sub>および OR ゲート 2 8<sub>-1</sub>により請求項 4 9 の第 1 の切替回路が構成される。

#### 【 0 0 8 0 】

2 段目の D 型フリップフロップ 2 2<sub>-1</sub>の出力データは、+ 8 倍の乗算器 2 9<sub>-1</sub>（請求項 4 9 の第 2 の乗算器に相当）を介して出力される。上記式(12)(13)に示すように、第 2 項については両式において土の符号が変わることはないので、第 1 項のようなクロック C K による符号切替のための回路は不要である。

**【 0 0 8 1 】**

また、3 段目の D 型フリップフロップ 2 3 \_1 の出力データは、- 1 倍の乗算器 3 0 \_1（請求項 4 9 の第 3 の乗算器に相当）を介して AND ゲート 3 1 \_1 の一方の入力端に入力されるとともに、乗算器を介することなく（+ 1 倍に相当）AND ゲート 3 2 \_1 の一方の入力端に入力される。AND ゲート 3 1 \_1 の他方の入力端には、クロック CK が入力される。また、AND ゲート 3 2 \_1 の他方の入力端には、インバータ 3 3 \_1 を通過した反転クロック CK が入力される。

**【 0 0 8 2 】**

上記 2 つの AND ゲート 3 1 \_1, 3 2 \_1 の出力データは、OR ゲート 3 4 \_1 を介して出力される。これにより、クロック CK が“H”の期間中は - 1 倍された間引きデータが OR ゲート 3 4 \_1 から出力される。また、クロック CK が“L”の期間中は + 1 倍された間引きデータが OR ゲート 3 4 \_1 から出力される。すなわち、クロック CK が“H”の期間中に上記式(12)に示す第 3 項が求められ、“L”の期間中に上記式(13)に示す第 3 項が求められる。2 つの AND ゲート 3 1 \_1, 3 2 \_1, インバータ 3 3 \_1 および OR ゲート 3 4 \_1 により請求項 4 9 の第 2 の切替回路が構成される。

**【 0 0 8 3 】**

上記 OR ゲート 2 8 \_1 の出力データ、8 倍乗算器 2 9 \_1 の出力データ、および OR ゲート 3 4 \_1 の出力データは、2 つの加算器 3 5 \_1, 3 6 \_1（請求項 4 9 の加算器に相当）によって全て加算される。これにより、クロック CK が“H”の期間中に上記式(12)に示す演算結果が加算器 3 6 \_1 より出力され、クロック CK が“L”の期間中に上記式(13)に示す演算結果が加算器 3 6 \_1 より出力される。

**【 0 0 8 4 】**

以上のような回路に対して、図 3（b）に示すような間引きデータ A, B, C, … を順に入力することにより、加算器 3 6 \_1 からは、A, B1, B2, C1, C2, … のように 2 倍にオーバーサンプリングされた補間データが出力される。なお、式(12)(13)ではデータ値が全体として 1 / 8 倍されているが、そのための回路は、1 段目のオーバーサンプリング回路の最終段に設けられている。

**【 0 0 8 5 】**

図 2 に示すデータ補間装置では、例えば 1 つの間引きデータ B から 2 つの補間データ B1, B2 を生成する際に、両補間データで共通している式 (12) (13) の第 2 項については、1 つの 8 倍乗算器 2 9 \_1 を共有してデータ形成するようにしている。このように、2 つの補間データを生成する際に共有できる回路はできる限り共有化することにより、全体としての回路構成を簡素化している。

#### 【0 0 8 6】

上記加算器 3 6 \_1 の出力段には、D 型フリップフロップ 3 7 \_1 と、加算器 3 8 \_1 と、1 / 1 6 倍乗算器 3 9 \_1 とから成る平均化回路が設けられている。なお、1 / 1 6 倍乗算器 3 9 \_1 の 1 / 8 倍分は、上記式 (12) (13) 中の 1 / 8 倍に相当し、残りの 1 / 2 倍分が平均化回路の一部を構成する。

#### 【0 0 8 7】

D 型フリップフロップ 3 7 \_1 は、基準周波数の 1 / 4 倍の周波数のクロック 2 C K に従って、加算器 3 6 \_1 の出力データを 1 クロック分遅延させる。加算器 3 8 \_1 は、加算器 3 6 \_1 の出力データと D 型フリップフロップ 3 7 \_1 の出力データとを加算する。1 / 1 6 倍乗算器 3 9 \_1 は、加算器 3 8 \_1 の出力データを 1 / 1 6 倍する。

#### 【0 0 8 8】

このように平均化回路を設けているのは、補間処理によって生じたクロックの相対位置のずれを元に戻すためである。例えば、加算器 3 6 \_1 から B1, B2, C1, C2, … のように補間データが出力されている場合、平均化回路では、 $(B1+B2)/2$ ,  $(B2+C1)/2$ ,  $(C1+C2)/2$ , … の演算を順に実行する。

#### 【0 0 8 9】

この平均化演算の実行結果は、

$$\begin{aligned} (B1+B2)/2 &= \{ \{ (A+8B-C)/8 \} + \{ (-A+8B+C)/8 \} \} / 2 = B \\ (B2+C1)/2 &= \{ \{ (-A+8B+C)/8 \} + \{ (B+8C-D)/8 \} \} / 2 \\ &= (-A+9B+9C-D)/16 \quad \cdots (14) \\ (C1+C2)/2 &= \{ \{ (B+8C-D)/8 \} + \{ (-B+8C+D)/8 \} \} / 2 = C \\ &\dots \end{aligned}$$

となり、クロックの相対位置のずれが元の正しい位置に修正される。このように

平均化演算を行うことにより、図 7 を用いて後述するように、得られる標本化関数の周波数特性を良好にすることが可能となる。

#### 【0 0 9 0】

上記  $1/16$  倍乗算器 3 9 \_1 の出力データは、2 段目のオーバーサンプリング回路の D 型フリップフロップ 2 1 \_2 に入力され、2 段目の 2 倍オーバーサンプリングが実行される。2 段目のオーバーサンプリング回路は、基準周波数の  $1/4$  倍の周波数のクロック 2 C K に従って動作すること以外は、1 段目のオーバーサンプリング回路と同様である。

#### 【0 0 9 1】

2 段目のオーバーサンプリング回路の最終段に設けられた  $1/16$  倍乗算器 3 9 \_2 の出力データは、3 段目のオーバーサンプリング回路の D 型フリップフロップ 2 1 \_3 に入力され、3 段目の 2 倍オーバーサンプリングが実行される。3 段目のオーバーサンプリング回路は、基準周波数の  $1/2$  倍の周波数のクロック 4 C K に従って動作すること以外は、1 段目のオーバーサンプリング回路と同様である。

#### 【0 0 9 2】

図 5 は、上記図 2 に示した伸長装置にユニットパルスのデータを入力した場合に得られるオーバーサンプル結果を示す特性図である。図 5 において、①はユニットパルスの入力データ、②は 2 倍オーバーサンプリングデータ、③は 4 倍オーバーサンプリングデータ、④は 8 倍オーバーサンプリングデータを示す。④に示すオーバーサンプリングデータの波形関数は、全域において 1 回以上微分可能であり、しかも有限の標本位置において 0 に収束する有限台の標本化関数である。

#### 【0 0 9 3】

したがって、本実施形態のデータ補間処理を適用した場合は、結果として図 5 の④に示す標本化関数を用いて離散的な各間引きデータに基づく重ね合わせを行うこととなり、間引きデータ間の値を 1 回以上微分可能な関数を用いて滑らかに補間することが可能となる。

#### 【0 0 9 4】

補間処理によって擬似的にサンプリング周波数を上げる従来のデータ補間方法

では、通常、 $\text{sinc}$ 関数と称される標本化関数が用いられる。

図6は、 $\text{sinc}$ 関数の説明図である。 $\text{sinc}$ 関数は、ディラックのデルタ関数を逆フーリエ変換したときに現れるものであり、標本化周波数を  $f$  としたときに  $\sin(\pi f t) / (\pi f t)$  で定義される。この  $\text{sinc}$ 関数は、 $t = 0$  のサンプル点のみで値が1になり、他の全てのサンプル点では値が0となる。

#### 【0095】

このような  $\text{sinc}$ 関数を用いたデータ補間方法では、求めた各補間値をサンプルホールド回路によって保持して階段状の信号波形を生成した後に、それをローパスフィルタに通すことによって滑らかな信号を出力するようにしていた。ところが、この方法では、出力される連続信号に、ローパスフィルタによる位相特性の劣化が生じるという問題があった。

#### 【0096】

また、上述した  $\text{sinc}$ 関数は、 $\pm\infty$ で0に収束する関数であるため、正確な補間値を求めようとすると、全てのサンプリングデータ値に対応した  $\text{sinc}$ 関数の値を求めて加算する必要がある。ところが実際には、処理能力や回路規模等の都合から、考慮するサンプリングデータの範囲を限定して畳み込み演算が行われていた。そのため、得られる補間値には打ち切り誤差が含まれ、正確な補間値が得られないという問題があった。

#### 【0097】

これに対して、図5④に示す標本化関数は、全域において1回以上微分可能であって、横軸に沿った標本位置が有限の局所的な領域内でのみ0以外の有限な値を有し、それ以外の領域では値が全て0となる有限台の関数である。

また、図5④の標本化関数は、中央のサンプル点のみで極大値をとり、他の幾つかのサンプル点で値が0になるという特徴を持つ関数であり、滑らかな波形の信号を得るために必要なサンプル点は全て通る。

#### 【0098】

したがって、図6に示す従来の  $\text{sinc}$ 関数の代わりに、図5④の標本化関数を用いて各間引きデータに基づく重ね合わせを行うことにより、間引きデータ間の値を1回以上微分可能な関数を用いて滑らかに補間することが可能となる。こ

れにより、ローパスフィルタを不要とし、その位相特性の劣化を防止することができる。

#### 【0099】

また、図5④の標本化関数は、有限のサンプル点で0に収束するため、その有限の範囲内での離散データだけを考慮に入れればよい。したがって、ある1つの補間値を求める場合には、限られた数の間引きデータの値のみを考慮すればよいことになり、処理量を大幅に削減することができる。しかも、その有限の範囲外の各間引きデータについては、本来考慮すべきであるが処理量や精度等を考慮して無視しているというわけではなく、理論的に考慮する必要がないため、打ち切り誤差の発生を防止することもできる。

#### 【0100】

また、圧縮側では、例えば2つのサンプリングデータB,Cから1つの間引きデータを求める際に、上記式(7)に示したように、対象のサンプリングデータB,Cだけを用いた単純な演算ではなく、その両隣のサンプリングデータA,Dも考慮に入れた演算により間引きデータを求めている。この場合の演算は、打ち切り誤差が発生しない標本化関数の元となるデジタル基本関数に基づいて行っているので、少なくとも両隣のサンプリングデータA,Dを考慮に入れて式(7)のように間引きデータを求めれば、伸長側で元データを忠実に再現するために必要な間引きデータを得ることができる。

#### 【0101】

図7は、図5④に示した標本化関数の周波数特性を示す図である。この図7に示すように、極めて良好な周波数特性が得られている。これによれば、いわゆる折り返しノイズが出るところにちょうどノッチフィルタが入ったような形になっており、折り返しノイズの発生を効果的に抑制することができる。

#### 【0102】

なお、上記図2に示した伸長装置は、本発明の伸長手法を実現するための回路構成の一例であって、これに限定されるものではない。

例えば、3つのD型フリップフロップの出力タップから取り出されるそれぞれのデータを-1倍、8倍、+1倍して加算する第1の乗加算回路と、3つのD型

フリップフロップの出力タップから取り出されるそれぞれのデータを+1倍、8倍、-1倍して加算する第2の乗加算回路と別個に設けるようにしても良い。

#### 【0103】

図8は、この場合の回路構成例を示す図である。なお、ここでは2倍のオーバーサンプルを行う回路のみを示しているが、同様の回路をn個カスケード接続することにより、 $2^n$ 倍のオーバーサンプルを行うことができるのは、図2の場合と同様である。また、ここでは平均化回路の図示を省略している。

#### 【0104】

図8において、3段のD型フリップフロップ41、42、43は、離散的に順次入力される間引きデータを1クロックCKずつ順に遅延させる。第1の乗加算回路は、1段目のD型フリップフロップ41の出力データを-1倍する第1の乗算器44と、2段目のD型フリップフロップ42の出力データを8倍する第2の乗算器45と、第1の乗算器44の出力データ、第2の乗算器45の出力データおよび3段目のD型フリップフロップ43の出力データを加算する加算器46、47とを備える。

#### 【0105】

また、第2の乗加算回路は、2段目のD型フリップフロップ42の出力データを8倍する第3の乗算器48と、3段目のD型フリップフロップ43の出力データを-1倍する第4の乗算器49と、第3の乗算器48の出力データ、第4の乗算器49の出力データおよび1段目のD型フリップフロップ41の出力データを加算する加算器50、51とを備える。

#### 【0106】

第1の乗加算回路（加算器47）から出力されたデータは、ANDゲート52の一方の入力端に入力される。ANDゲート52の他方の入力端には、インバータ53を通過した反転クロックCKが入力される。また、第2の乗加算回路（加算器51）から出力されたデータは、ANDゲート54の一方の入力端に入力される。ANDゲート54の他方の入力端には、クロックCKが入力される。

#### 【0107】

上記2つのANDゲート52、54の出力データは、ORゲート55を介して



出力される。これにより、クロック C K が “H” の期間中は、上記式(12)に示すような演算結果が O R ゲート 5 5 より出力され、クロック C K が “L” の期間中は、上記式(13)に示すような演算結果が O R ゲート 5 5 より出力される。

#### 【0 1 0 8】

また、図 9 に示すように、図 8 では 2 つ用いていた 8 倍乗算器を 1 つの乗算器で兼用するようにしても良い。なお、図 9 でも 2 倍のオーバーサンプルを行う回路のみを示しているが、同様の回路を  $n$  個カスケード接続することにより、 $2^n$  倍のオーバーサンプルを行うことができる。また、ここでも平均化回路の図示は省略している。

#### 【0 1 0 9】

図 9 に示す伸長装置は、順次入力される間引きデータを 1 クロック C K ずつ順に遅延させる 3 段の D 型フリップフロップ 6 1, 6 2, 6 3 を備えている。また、1 段目の D 型フリップフロップ 6 1 の出力データを -1 倍する第 1 の乗算器 6 4 と、2 段目の D 型フリップフロップ 6 2 の出力データを 8 倍する第 2 の乗算器 6 5 と、3 段目の D 型フリップフロップ 6 3 の出力データを -1 倍する第 3 の乗算器 6 6 とを備えている。

#### 【0 1 1 0】

さらに、第 1 の乗算器 6 4 の出力データ、第 2 の乗算器 6 5 の出力データおよび 3 段目の D 型フリップフロップ 6 3 の出力データを加算する第 1 の加算器 6 7, 6 8 と、第 2 の乗算器 6 5 の出力データ、第 3 の乗算器 6 6 の出力データおよび 1 段目の D 型フリップフロップ 6 1 の出力データを加算する第 2 の加算器 6 9, 7 0 とを備えている。

#### 【0 1 1 1】

加算器 6 8 から出力されたデータは、AND ゲート 7 1 の一方の入力端に入力される。AND ゲート 7 1 の他方の入力端には、インバータ 7 2 を通過した反転クロック C K が入力される。また、加算器 7 0 から出力されたデータは、AND ゲート 7 3 の一方の入力端に入力される。AND ゲート 7 3 の他方の入力端には、クロック C K が入力される。

#### 【0 1 1 2】

上記2つのANDゲート71, 73の出力データは、ORゲート74を介して出力される。これにより、クロックCKが“H”の期間中は、上記式(12)に示すような演算結果がORゲート74より出力され、クロックCKが“L”の期間中は、上記式(13)に示すような演算結果がORゲート74より出力される。

#### 【0113】

また、上記実施形態では、上記式(12)(13)のような演算を行うことによってデータ補間を行い、求めた補間データに対して更に平均化演算を行うことによってクロックの相対位置のずれを補正するようにしていた。これに対して、上記式(14)に示すような演算を直接行うことによって、平均化演算を省略できるようにしても良い。

#### 【0114】

上記式(14)に示すような演算を直接行う場合は、順次入力される間引きデータについて、注目する隣接する2つのサンプル点の間引きデータとその両隣のサンプル点の間引きデータとに対して、デジタル基本関数の値に応じた重み付け加算をすることにより、上記2つのサンプル点の間引きデータの間を補間する補間データを順次求めることになる。例えば、注目する2つの間引きデータがB,Cの場合は、その間の補間データ  $(B+C)/2$  は、次式のように求められる。

$$(B+C)/2 = (-A+9B+9C-D)/16 \quad \cdots(15)$$

#### 【0115】

図10は、この場合の回路構成例を示す図である。なお、ここでも2倍のオーバーサンプリングを行う回路のみを示しているが、同様の回路をn個カスケード接続することにより、 $2^n$ 倍のオーバーサンプリングを行うことが可能である。

#### 【0116】

図10に示す伸長装置は、順次入力される間引きデータを1クロックCKずつ順に遅延させる4段のD型フリップフロップ81, 82, 83, 84を備えている。この4段のD型フリップフロップ81～84は、請求項55に記載した4段の遅延回路に相当する。

#### 【0117】

また、1段目のD型フリップフロップ81の出力データを-1倍する第1の乗

算器 8 5 と、2 段目の D 型フリップフロップ 8 2 の出力データを 9 倍する第 2 の乗算器 8 6 と、3 段目の D 型フリップフロップ 8 3 の出力データを 9 倍する第 3 の乗算器 8 7 と、4 段目の D 型フリップフロップ 8 4 の出力データを - 1 倍する第 4 の乗算器 8 8 とを備えている。さらに、上記第 1 ～第 4 の乗算器 8 5 ～ 8 8 の出力データを全て加算する加算器 8 9, 9 0, 9 1 を備えている。

#### 【0 1 1 8】

加算器 9 1 から出力されたデータは、AND ゲート 9 2 の一方の入力端に入力される。AND ゲート 9 2 の他方の入力端には、インバータ 9 3 を通過した反転クロック C K が入力される。また、初段の D 型フリップフロップ 8 1 に入力される間引きデータは、上述の各回路ブロック 8 1 ～ 9 1 の遅延に合わせた量だけ遅延させる遅延回路 9 4 を介して AND ゲート 9 5 の一方の入力端に入力される。AND ゲート 9 5 の他方の入力端には、クロック C K が入力される。

#### 【0 1 1 9】

上記 2 つの AND ゲート 9 2, 9 5 の出力データは、OR ゲート 9 6 を介して出力される。これにより、クロック C K が “H” の期間中は、入力される間引きデータがそのまま OR ゲート 9 6 より出力され、クロック C K が “L” の期間中は、上記式 (15) に示すような演算により得られる補間データが OR ゲート 9 6 より出力される。

#### 【0 1 2 0】

なお、この図 1 0 は上記式 (15) のような演算を行うための 1 つの例であり、これに限定されるものではない。例えば、図 1 0 に示した乗加算回路 8 5 ～ 9 1 の代わりに、以下のような構成を用いても良い。すなわち、1 段目の D 型フリップフロップ 8 1 の出力データと 4 段目の D 型フリップフロップ 8 4 の出力データとを加算する第 1 の加算器と、2 段目の D 型フリップフロップ 8 2 の出力データと 3 段目の D 型フリップフロップ 8 3 の出力データとを加算する第 2 の加算器と、上記第 1 の加算器の出力データを - 1 倍する第 1 の乗算器と、上記第 2 の加算器の出力データを 9 倍する第 2 の乗算器と、上記第 1 の乗算器の出力データと上記第 2 の乗算器の出力データとを加算する第 3 の加算器とにより乗加算回路を構成する。このようにすれば、乗算器を削減することができる。

**【0121】**

以上詳しく説明したように、第1の実施形態によれば、圧縮側において、全域において1回以上微分可能な有限台の標本化関数の元となるデジタル基本関数に基づいて間引きデータを求めることにより、8倍の圧縮率を達成することができる。また、伸長側において同じデジタル基本関数を用いた補間データを求めることにより、圧縮前の元データをほぼ忠実に再現することができる。その際の間引きデータは上記式(7)のような極めて簡単な四則演算のみで求めることができ、補間データは上記式(12)(13)、あるいは式(14)のような極めて簡単な四則演算のみで求めることができる。

**【0122】**

これらの式(7)(12)(13)(14)によれば、あるサンプル点の離散データについて間引きデータや補間データを求める際には、注目するサンプル点の離散データとその周辺数個のサンプル点の離散データのみを考慮に入れば良い。また、圧縮対象のデータを時間／周波数変換することなく、時間軸上でそのまま圧縮・伸長することができるので、処理が複雑にならず、構成を簡素化することもできる。

**【0123】**

したがって、本実施形態によれば、高圧縮率化と再生データの高品質化との両方を実現することができるばかりでなく、そのための演算時間を短縮化することができるとともに、その演算回路を簡素化することができる。

**【0124】**

なお、上記第1の実施形態では、デジタル基本関数として、1クロック毎にデータ値を-1, 1, 8, 8, 1, -1と変化させたものを用いた。この数値が最良ではあるが、これ以外の数値を用いたもの（例えば、両脇に当たる部分のウェイトを-1ではなく1あるいは0にしたもの、真中に当たる部分のウェイトを8以外の値にしたものなど）をデジタル基本関数としても良い。

**【0125】**

また、上記第1の実施形態では、圧縮装置および伸長装置をロジック回路を組み合わせてハードウェア的に構成しているが、CPUあるいはMPU、ROM、RAMなどを備えたコンピュータシステムによって構成し、上述の圧縮装置およ

び伸長装置の機能を上述の R O M や R A M など に格納されたプログラムが動作することによって実現することも可能である。

#### 【0 1 2 6】

(第 2 の実施形態)

次に、本発明の第 2 の実施形態について説明する。第 2 の実施形態では、図 4 に示すデジタル基本関数を 1 クロックずつずらして加算 (平均化) した関数を使用する。

#### 【0 1 2 7】

図 1 1 は、第 2 の実施形態による間引きおよび補間の原理を説明するための図である。図 1 1 (a) は間引きの原理を示し、図 1 1 (b) は補間の原理を示している。まず、図 1 1 (a) を用いて本実施形態による間引きの原理について説明する。

#### 【0 1 2 8】

図 1 1 (a) に示すデータ構造において、横軸の A, B, C, … は 1 クロック毎に順次入力される各サンプル点のサンプリングデータを示し、そのデータ値が本実施形態のデジタル基本関数に対する係数となる。また、縦軸の a, b, c, … は、デジタル基本関数により加工されたサンプリングデータの中心位置を示す。

#### 【0 1 2 9】

図 1 1 (a) に示すように、サンプリングデータ A に対するデジタル基本関数は、最初のクロック位置を先頭として各関数値 (−1, 0, 9, 16, 9, 0, −1) を 1 クロックずつずらして与えられる。また、次のサンプリングデータ B に対するデジタル基本関数は、最初のクロック位置より 1 クロック後のクロック位置を先頭として各関数値を 1 クロックずつずらして与えられる。同様に、順次入力される各サンプリングデータ C, D, E, F, … に対するデジタル基本関数は、更に 1 クロックずつ遅れたクロック位置をそれぞれ先頭として、各関数値を 1 クロックずつずらして与えられる。

#### 【0 1 3 0】

第 2 の実施形態では、このようなデータ構造に基づいて、各サンプル点のサンプリングデータ A, B, C, … を  $1/2$  に間引く演算を行う。第 2 の実施形態において

は、デジタル基本関数の中心位置が第 1 の実施形態と異なり 1 つのデータ位置の上に重なる。そこで、連続するサンプリングデータを 1 つ飛びに採用し、採用したサンプル点については、それに対応するデジタル基本関数の中心位置の行においてデータ値を畳み込み加算した値に置き換える処理を行う。

#### 【0 1 3 1】

すなわち、最初の 3 つのサンプリングデータ A, B, C については、演算に必要なデータが不足していることから、間引きを行うことなくそのままの状態とする。次のサンプリングデータ D については、それに対応するデジタル基本関数の中心位置 d の行において、次式(16)のように各データ値を加算して振幅分を調整した結果を間引きデータとする。

$$D \rightarrow (16D + 9(C + E) - (A + G)) / 32 \quad \cdots (16)$$

#### 【0 1 3 2】

また、次のサンプリングデータ E は破棄して更に次のサンプリングデータ F を採用し、これを、対応するデジタル基本関数の中心位置 f の行において次式(17)のように演算した間引きデータに置き換える。

$$F \rightarrow (16F + 9(E + G) - (C + I)) / 32 \quad \cdots (17)$$

以降同様に、1 つ飛びのクロック位置 h, j, l, … における畳み込み加算値を間引きデータとして得る。

#### 【0 1 3 3】

このようにして 1 / 2 に間引いたデータをもとに同様の間引き演算を行うことにより、元データを 1 / 4 に間引くことが可能である。さらに、1 / 4 に間引いたデータをもとに同様の間引き演算を行うことにより、元データを 1 / 8 に間引くことが可能である。理論的には、同様の間引き演算を n 回繰り返すことにより、元データを 1 / 2<sup>n</sup> 倍に圧縮することが可能である。

#### 【0 1 3 4】

次に、図 1 1 (b) を用いて第 2 の実施形態によるデータ補間の原理について説明する。図 1 1 (b) に示すデータ構造において、横軸の A, B, C, … は 1 クロック毎に順次入力される間引きデータを示し、そのデータ値が本実施形態のデジタル基本関数に対する係数となる。また、縦軸の a, b, c, … は、デジタル基本関数に

より加工された間引きデータの中心位置を示す。

### 【 0 1 3 5 】

図 1 1 ( b ) に示すように、間引きデータ A に対するデジタル基本関数は、最初のクロック位置より 2 クロック前を先頭として各関数値 ( - 1 , 0 , 9 , 1 6 , 9 , 0 , - 1 ) を 1 クロックずつずらして与えられる。また、次の間引きデータ B に対するデジタル基本関数は、最初のクロック位置を先頭として各関数値を 1 クロックずつずらして与えられる。同様に、順次入力される各間引きデータ C , D , E , F , … に対するデジタル基本関数は、更に 2 クロックずつ遅れたクロック位置をそれぞれ先頭として、各関数値を 1 クロックずつずらして与えられる。

### 【 0 1 3 6 】

このようなデータ構造によると、例えば間引きデータ C の値に応じて加工されるデジタル基本関数の中心位置 c は、第 1 の実施形態と異なり 1 つのデータ位置の上に重なる。そこで、第 2 の実施形態では、このクロック位置 c のデータと、その前後のクロック位置のデータとを用いて畳み込み演算を行うことにより、1 つの間引きデータ C から 2 つの補間データ C1 , C2 を求める。

### 【 0 1 3 7 】

クロック位置 c を中心とする前後 3 行分のデータをそれぞれ、c- , c , c+ とすると、

$$c- = -A + 9B + 9C - D \quad \cdots (18)$$

$$c = 16C \quad \cdots (19)$$

$$c+ = -B + 9C + 9D - E \quad \cdots (20)$$

と表される。

### 【 0 1 3 8 】

上記式 (18) (19) より、

$$(c- + c) = (-A + 9B + 25C - D) \quad \cdots (21)$$

と表され、また、上記式 (19) (20) より、

$$(c + c+) = (-B + 25C + 9D - E) \quad \cdots (22)$$

と表される。

### 【 0 1 3 9 】

この式(21)(22)をもとに振幅の調整分を考慮すると、1つの間引きデータCは、2つの補間データC1, C2に置き換えることができ、それらは、

$$C1 = (-A + 9B + 25C - D) / 32 \quad \cdots (23)$$

$$C2 = (-B + 25C + 9D - E) / 32 \quad \cdots (24)$$

のように表される。

#### 【0 1 4 0】

また、間引きデータCについて式(23)(24)に示される関係が他の間引きデータD, E, F, …についても同様に成り立つ。したがって、次式に示すように、1つの間引きデータDは2つの補間データD1, D2に、1つの間引きデータEは2つの補間データE1, E2に、1つの間引きデータFは2つの補間データF1, F2にそれぞれ置き換えられる。

$$D \rightarrow D1 = (-B + 9C + 25D - E) / 32, D2 = (-C + 25D + 9E - F) / 32$$

$$E \rightarrow E1 = (-C + 9D + 25E - F) / 32, E2 = (-D + 25E + 9F - G) / 32$$

$$F \rightarrow F1 = (-D + 9E + 25F - G) / 32, F2 = (-E + 25F + 9G - H) / 32$$

.....

#### 【0 1 4 1】

以上のように、あるサンプル点の間引きデータ（例えばC）に関して補間値を求める際には、その注目するサンプル点の間引きデータを25倍した値と、1つ前のサンプル点の間引きデータを9倍した値と、2つ前のサンプル点の間引きデータを-1倍した値と、1つ後のサンプル点の間引きデータを-1倍した値とを加算し、その加算値を32で割ることによって第1の補間値（例えばC1）を得る。

#### 【0 1 4 2】

また、注目するサンプル点の間引きデータを25倍した値と、1つ前のサンプル点の間引きデータを-1倍した値と、1つ後のサンプル点の間引きデータを9倍した値と、2つ後のサンプル点の間引きデータを-1倍した値とを加算し、その加算値を32で割ることによって第2の補間値（例えばC2）を得る。

これを各サンプル点について行うことにより、元データを2倍にオーバーサンプルする。



**【 0 1 4 3 】**

また、2 倍にオーバーサンプルされた補間データをもとに同様の補間処理を行うことにより、元データを 4 倍にオーバーサンプルすることが可能である。さらに、4 倍にオーバーサンプルされた補間データをもとに同様の補間処理を行うことにより、元データを 8 倍にオーバーサンプルすることが可能である。理論的には、同様の補間処理を  $n$  回繰り返し行うことにより、元データを  $2^n$  倍にオーバーサンプルすることが可能である。

**【 0 1 4 4 】**

図 1 2 は、上述の間引き動作を実現する第 2 の実施形態による圧縮装置（間引き装置）の構成例を示す図である。

図 1 2 に示す圧縮装置は、 $1/2$  倍の間引きを行うための回路を 3 段カスケード接続した構成になっており、1 段目～3 段目の回路構成を符号の添数字（-1～-3）により区別している。また、添数字のみが異なり主符号が同じ回路構成は互いに同一の機能を有するものであることを示している。

**【 0 1 4 5 】**

1 段目の間引き回路は、基準周波数（例えば 4 4 . 1 K H z）のクロック 8 C K に従って動作する。7 段の D 型フリップフロップ 1 0 1 \_1, 1 0 2 \_1, 1 0 3 \_1, 1 0 4 \_1, 1 0 5 \_1, 1 0 6 \_1, 1 0 7 \_1 は、離散的に順次入力されるサンプリングデータ（例えば 1 6 ビット）を基準周波数の 1 クロック 8 C K ずつ順に遅延させる。これらの D 型フリップフロップ 1 0 1 \_1～1 0 7 \_1 は、請求項 5 に記載した 7 段の遅延回路に相当する。

**【 0 1 4 6 】**

加算器 1 0 8 \_1 は、1 段目と 7 段目の D 型フリップフロップ 1 0 1 \_1, 1 0 7 \_1 の出力タップから取り出されたデータを加算する。乗算器 1 0 9 \_1 は、上記加算器 1 0 8 \_1 の出力データを - 1 倍する。上記加算器 1 0 8 \_1 および乗算器 1 0 9 \_1 は、請求項 6 に記載した第 1 の乗加算回路に相当する。

**【 0 1 4 7 】**

加算器 1 1 0 \_1 は、3 段目と 5 段目の D 型フリップフロップ 1 0 3 \_1, 1 0 5 \_1 の出力タップから取り出されたデータを加算する。乗算器 1 1 1 \_1 は、上記加

算器 1 1 0<sub>-1</sub> の出力データを 9 倍する。上記加算器 1 1 0<sub>-1</sub> および乗算器 1 1 1<sub>-1</sub> は、請求項 6 に記載した第 2 の乗加算回路に相当する。

#### 【0 1 4 8】

乗算器 1 1 2<sub>-1</sub> は、4 段目の D 型フリップフロップ 1 0 4 の出力タップから取り出されたデータを 1 6 倍する。

加算器 1 1 3<sub>-1</sub>, 1 1 4<sub>-1</sub> は、上記 3 つの乗算器 1 0 9<sub>-1</sub>, 1 1 1<sub>-1</sub>, 1 1 2<sub>-1</sub> の出力データを加算する。乗算器 1 1 5<sub>-1</sub> は、上記加算器 1 1 4<sub>-1</sub> の出力データを 1 / 3 2 倍する。上記加算器 1 1 3<sub>-1</sub>, 1 1 4<sub>-1</sub> および乗算器 1 1 5<sub>-1</sub> は、請求項 6 に記載した第 3 の乗加算回路に相当する。

#### 【0 1 4 9】

以上のような回路に対して、図 1 1 (a) に示すようなサンプリングデータ A, B, C, … を順に入力することにより、乗算器 1 1 5<sub>-1</sub> からは、式 (16) (17) のように示される 1 / 2 間引きデータが出力される。乗算器 1 1 5<sub>-1</sub> の出力データは、2 段目の間引き回路の D 型フリップフロップ 1 0 1<sub>-2</sub> に入力され、2 段目の 1 / 2 間引きが実行される。2 段目の間引き回路は、基準周波数の 1 / 2 倍の周波数のクロック 4 C K に従って動作すること以外は、1 段目の間引き回路と同様である。

#### 【0 1 5 0】

なお、図 1 2 の回路によれば、サンプリングデータ D に関する式 (16) に示す間引きデータが乗算器 1 1 5<sub>-1</sub> から出力された後は、サンプリングデータ E に関する間引きデータが出力され、その後でサンプリングデータ F に関する式 (17) に示す間引きデータが出力される。しかし、サンプリングデータ E に関する間引きデータは、2 段目の間引き回路が基準周波数の 1 / 2 倍の周波数のクロック 4 C K に従って動作しているために無視され、次のサンプリングデータ F に関する間引きデータが引き続き 2 段目の間引き回路で処理される。

#### 【0 1 5 1】

2 段目の間引き回路の最終段に設けられた乗算器 1 1 5<sub>-2</sub> の出力データは、3 段目の間引き回路の D 型フリップフロップ 1 0 1<sub>-3</sub> に入力され、3 段目の 1 / 2 間引きが実行される。3 段目の間引き回路は、基準周波数の 1 / 4 倍の周波数の

クロック 2 C Kに従って動作すること以外は、1 段目の間引き回路と同様である。3 段目の間引き回路より出力されたデータは、基準周波数の  $1/8$  倍の周波数のクロック C Kに従って D 型フリップフロップ 1 1 6 に保持された後、最終的な間引きデータ（圧縮データ）として出力される。

#### 【0 1 5 2】

図 1 3 は、第 2 の実施形態による伸長装置（補間装置）の構成例を示す図である。図 1 3 に示す伸長装置は、2 倍のオーバーサンプリングを行うための回路を 3 段カスケード接続した構成になっており、1 段目～3 段目の回路構成を符号の添数字（-1～-3）により区別している。また、添数字のみが異なり主符号が同じ回路構成は互いに同一の機能を有するものであることを示している。

#### 【0 1 5 3】

1 段目のオーバーサンプリング回路は、基準周波数の  $1/8$  倍の周波数の（5 1 2 5 K H z）のクロック C Kに従って動作する。5 段の D 型フリップフロップ 1 2 1<sub>-1</sub>, 1 2 2<sub>-1</sub>, 1 2 3<sub>-1</sub>, 1 2 4<sub>-1</sub>, 1 2 5<sub>-1</sub> は、順次入力される間引きデータ（例えば 1 6 ビット）を 1 クロック C K ずつ順に遅延させる。これらの D 型フリップフロップ 1 2 1<sub>-1</sub>～1 2 5<sub>-1</sub> は、請求項 5 9 による 5 段の遅延回路に相当する。

#### 【0 1 5 4】

1 段目の D 型フリップフロップ 1 2 1<sub>-1</sub> の出力タップから取り出されたデータは、-1 倍の乗算器 1 2 6<sub>-1</sub> に入力される。2 段目の D 型フリップフロップ 1 2 2<sub>-1</sub> の出力タップから取り出されたデータは、+9 倍の乗算器 1 2 7<sub>-1</sub> と -1 倍の乗算器 1 3 3<sub>-1</sub> とに入力される。

#### 【0 1 5 5】

3 段目の D 型フリップフロップ 1 2 3<sub>-1</sub> の出力タップから取り出されたデータは、2 5 倍の乗算器 1 2 8<sub>-1</sub> に入力される。4 段目の D 型フリップフロップ 1 2 4<sub>-1</sub> の出力タップから取り出されたデータは、-1 倍の乗算器 1 2 9<sub>-1</sub> と +9 倍の乗算器 1 3 4<sub>-1</sub> とに入力される。5 段目の D 型フリップフロップ 1 2 5<sub>-1</sub> の出力タップから取り出されたデータは、-1 倍の乗算器 1 3 5<sub>-1</sub> に入力される。

#### 【0 1 5 6】

上記－1 倍乗算器 1 2 6<sub>-1</sub>， 9 倍乗算器 1 2 7<sub>-1</sub>， 2 5 倍乗算器 1 2 8<sub>-1</sub>および－1 倍乗算器 1 2 9<sub>-1</sub>の出力データは、3つの加算器 1 3 0<sub>-1</sub>， 1 3 1<sub>-1</sub>， 1 3 2<sub>-1</sub>によって全て加算される。上記－1 倍乗算器 1 3 3<sub>-1</sub>， 2 5 倍乗算器 1 2 8<sub>-1</sub>， 9 倍乗算器 1 3 4<sub>-1</sub>および－1 倍乗算器 1 3 5<sub>-1</sub>の出力データは、3つの加算器 1 3 6<sub>-1</sub>， 1 3 7<sub>-1</sub>， 1 3 8<sub>-1</sub>によって全て加算される。

#### 【0 1 5 7】

加算器 1 3 2<sub>-1</sub>から出力されたデータは、ANDゲート 1 3 9<sub>-1</sub>の一方の入力端に入力される。ANDゲート 1 3 9<sub>-1</sub>の他方の入力端には、クロック C Kが入力される。また、加算器 1 3 8<sub>-1</sub>から出力されたデータは、ANDゲート 1 4 0<sub>-1</sub>の一方の入力端に入力される。ANDゲート 1 4 0<sub>-1</sub>の他方の入力端には、インバータ 1 4 1<sub>-1</sub>を通過した反転クロック C Kが入力される。

#### 【0 1 5 8】

上記2つのANDゲート 1 3 9<sub>-1</sub>， 1 4 0<sub>-1</sub>の出力データは、ORゲート 1 4 2<sub>-1</sub>を介して1／3 2 倍の乗算器 1 4 3<sub>-1</sub>に供給される。これにより、クロック C Kが“H”の期間中は、上記式(23)に示すような演算結果が1／3 2 倍乗算器 1 4 3<sub>-1</sub>より出力され、クロック C Kが“L”の期間中は、上記式(24)に示すような演算結果が1／3 2 倍乗算器 1 4 3<sub>-1</sub>より出力される。

#### 【0 1 5 9】

以上のような回路に対して、図 1 1 (b)に示すような間引きデータA,B,C, …を順に入力することにより、1／3 2 倍乗算器 1 4 3<sub>-1</sub>からは、A,B,C1,C2,D1,D2, …のように2 倍にオーバーサンプリングされた補間データが出力される。

#### 【0 1 6 0】

上記1／3 2 倍乗算器 1 4 3<sub>-1</sub>の出力データは、2 段目のオーバーサンプリング回路のD型フリップフロップ 1 2 1<sub>-2</sub>に入力され、2 段目の2 倍オーバーサンプリングが実行される。2 段目のオーバーサンプリング回路は、基準周波数の1／4 倍の周波数のクロック 2 C Kに従って動作すること以外は、1 段目のオーバーサンプリング回路と同様である。

#### 【0 1 6 1】

2 段目のオーバーサンプリング回路の最終段に設けられた1／3 2 倍乗算器 1

4 3 \_2 の出力データは、3 段目のオーバーサンプリング回路の D 型フリップフロップ 1 2 1 \_3 に入力され、3 段目の 2 倍オーバーサンプリングが実行される。3 段目のオーバーサンプリング回路は、基準周波数の  $1/2$  倍の周波数のクロック 4 C K に従って動作すること以外は、1 段目のオーバーサンプリング回路と同様である。

#### 【0 1 6 2】

3 段目のオーバーサンプリング回路の最終段に設けられた  $1/32$  倍乗算器 1 4 3 \_3 の出力データは、基準周波数のクロック 8 C K に従って D 型フリップフロップ 1 4 4 に保持された後、最終的な補間データ（伸長データ）として出力される。

#### 【0 1 6 3】

この図 1 3 のように伸長装置を構成した場合も、ユニットパルスのデータを入力した場合に得られる標本化関数は、図 5 ④ に示したものとほぼ同じものとなる。したがって、第 2 の実施形態による補間処理を適用した場合も、間引きデータ間の値を 1 回以上微分可能な関数を用いて滑らかに補間することが可能となる。これにより、ローパスフィルタを不要とし、その位相特性の劣化を防止することができる。

#### 【0 1 6 4】

また、ある 1 つの間引きデータに関して補間値を求める場合には、限られた数（上記式 (23) (24) で示されるように 4 個）の間引きデータの値のみを考慮すればよいことになり、処理量を大幅に削減することができる。しかも、有限台の範囲外の各間引きデータについては、本来考慮すべきであるが処理量や精度等を考慮して無視しているというわけではなく、理論的に考慮する必要がないため、打ち切り誤差の発生を防止することもできる。

#### 【0 1 6 5】

以上詳しく説明したように、第 2 の実施形態によれば、第 1 の実施形態と同様に、高圧縮率化と再生データの高品質化との両方を実現することができるだけでなく、そのための演算時間を短縮化することができるとともに、その演算回路を簡素化することができる。

## 【0166】

また、第2の実施形態では、図11に示すように、図4に示すデジタル基本関数を1クロックずつずらして加算（平均化）した関数を使用して補間データを求めているので、この関数内で既に平均化演算は行われている。したがって、上記式(23)(24)による補間演算を行った後に平均化演算を行う必要がなく、平均化回路を省略することができる。

## 【0167】

なお、上記第2の実施形態でも、デジタル基本関数として、1クロック毎にデータ値を-1, 1, 8, 8, 1, -1と変化させたものを用いた。この数値が最良ではあるが、これ以外の数値を用いたもの（例えば、両脇に当たる部分のウェイトを-1ではなく1あるいは0にしたもの、真中に当たる部分のウェイトを8以外の値にしたものなど）をデジタル基本関数としても良い。

## 【0168】

また、上記第2の実施形態でも、圧縮装置および伸長装置をロジック回路を組み合わせてハードウェア的に構成しているが、CPUあるいはMPU、ROM、RAMなどを備えたコンピュータシステムによって構成し、上述の圧縮装置および伸長装置の機能を上述のROMやRAMなどに格納されたプログラムが動作することによって実現することも可能である。

## 【0169】

（第3の実施形態）

次に、本発明の第3の実施形態について説明する。

図14は、第3の実施形態による圧縮装置の構成例を示す図である。図14に示すように、本実施形態の圧縮装置は、無音処理部201、間引き処理部202、直線圧縮部203、丸め演算部204およびブロック化部205を備えて構成されている。

## 【0170】

無音処理部201は、本発明の置換手段に相当するものであり、圧縮対象として入力される各サンプリングデータの絶対値を第1の所定値（例えば“4”）だけ小さく丸める処理を行う。また、入力されるサンプリングデータの絶対値が上

記第1の所定値よりも小さいか、入力されるサンプリングデータの絶対値が第2の所定値（例えば“16”）よりも小さく、かつ、微分データの短区間平均値（短区間のデフォルト値は例えば8クロック）が第3の所定値（例えば“8”）よりも小さい場合には、そのサンプリングデータを無音とみなして、データ値を“0”に置き換えて出力する。

#### 【0171】

このような処理を行うことによって、DCオフセットを修正して入力データ中に含まれる細かな雑音成分を除去するとともに、圧縮率の向上を図っている。

#### 【0172】

間引き処理部202は、本発明の間引き手段に相当するものであり、無音処理部201より出力されたサンプリングデータに対し、第1の実施形態あるいは第2の実施形態で説明したような何れかの間引き処理を行う。何れかの間引き処理を選択的に切り替えて適用することができるようにしても良い。圧縮対象のデータをこの間引き処理部202に通すことにより、最低でも8倍の圧縮率を実現できる。

#### 【0173】

直線圧縮部203は、本発明の標本点検出手段および圧縮データ生成手段を含み、間引き処理部202より出力された間引きデータに対して、以下に説明するような直線圧縮の処理を行う。すなわち、間引き処理部202より出力された各間引きデータのサンプル点の中から、2つの間引きデータ間を結ぶ直線上のデータ値と、その直線上のデータ値と同じサンプル点における間引きデータ値との誤差が所望の値以下となるサンプル点を標本点として順次検出する。そして、検出した各標本点における離散的な振幅データ値と、各標本点間の時間間隔を表すタイミングデータ値とを求める。

#### 【0174】

上記標本点を検出する処理をより具体的に説明すると、以下の通りである。すなわち、間引きデータ中から、基準とする間引きデータと、そこからの時間間隔が所定範囲内にあるもう一方の間引きデータとを選ぶ。そして、その2つの間引きデータ間を結ぶ直線上の各データ値と、その直線上の各データ値と同じサン

ル点における各間引きデータ値との誤差が全て所望の値以下となるサンプル点であって、上記所定範囲の中で時間間隔が最も長くなるサンプル点を標本点として検出する。

#### 【0 1 7 5】

図 1 5 は、直線圧縮の基本原理を説明するための図である。図 1 5 において、横軸は時間を表し、縦軸は間引きデータの振幅を表す。図 1 5 中に示す D 1 ～ D 9 は、1 クロック C K のサンプリング周期で間引き処理部 2 0 2 から直線圧縮部 2 0 3 に入力される間引きデータの一部である。

#### 【0 1 7 6】

この図 1 5 の例では、間引きデータ D 1 を最初に採用する基準の間引きデータとしている。また、標本点を検出する際に選ぶ 2 つの間引きデータ間の時間間隔は、最大で 6 クロックの範囲としている。なお、タイミングデータ値として 3 ビットあるいは 4 ビットを用いる場合、間引きデータ間の時間間隔は最大で 7 クロックあるいは 1 5 クロックとすることが可能である。

#### 【0 1 7 7】

まず、図 1 5 (a) に示すように、基準の間引きデータ D 1 と、そこから時間間隔が所定範囲内で最大となる間引きデータ D 7 とを選ぶ。そして、その 2 つの間引きデータ間を結ぶ直線上にある各サンプル点のデータ値 D 2' , D 3' , D 4' , D 5' , D 6' と、その直線上の各データ値 D 2' ～ D 6' と同じサンプル点における各間引きデータ値 D 2 , D 3 , D 4 , D 5 , D 6 とのそれぞれの誤差が、全て所望の値以下となるかどうかを判断する。

#### 【0 1 7 8】

すなわち、2 つの間引きデータ D 1 - D 7 間を結ぶ直線上の各データ値 D 2' , D 3' , D 4' , D 5' , D 6' と各間引きデータ値 D 2 , D 3 , D 4 , D 5 , D 6 との誤差の全てが、点線で示す所望の値の範囲内にあるかどうかを判断する。この条件を満たす場合には、間引きデータ D 7 のサンプル点を標本点として検出する。しかし、この例では、直線上のデータ値 D 4' とそれに対応する間引きデータ値 D 4 との誤差が所望の値を超えているので、この時点では間引きデータ D 7 のサンプル点を標本点としては採用せず、処理を先に進める。



**【0179】**

次に、図15(b)に示すように、基準の間引きデータD1からの時間間隔が間引きデータD7よりも1クロックCK短い間引きデータD6を選ぶ。そして、2つの間引きデータD1-D6間を結ぶ直線上にある各サンプル点のデータ値D2'', D3'', D4'', D5''と、その直線上の各データ値D2''~D5''と同じサンプル点における各間引きデータ値D2, D3, D4, D5とのそれぞれの誤差が、全て所望の値以下となるかどうかを判断する。

**【0180】**

そして、全ての誤差が所望の値以下となる場合には、間引きデータD6のサンプル点を標本点として検出する。この例では、直線上の各データ値D2'', D3'', D4'', D5''と各間引きデータ値D2, D3, D4, D5との誤差が全て所望の値以下となるので、この間引きデータD6のサンプル点を標本点として検出する。

**【0181】**

なお、D1-D7間、D1-D6間、…、D1-D3間に結んだそれぞれの直線に関して、全ての誤差が所望の値以下になるという誤差の条件を何れも満たさなかった場合は、間引きデータD2のサンプル点を標本点として検出する。すなわち、間引きデータD1-D2間には他の間引きデータが存在しないので、この区間については上述の誤差演算を行う必要がない。よって、他の区間に結んだそれぞれの直線に関して誤差の条件を何れも満たさなかった場合には、現在基準としている間引きデータD1の隣りの間引きデータD2の位置を標本点として検出する。

**【0182】**

1つの標本点を検出したら、その標本点のデータを新たに基準の間引きデータとして用い、そこから6クロックの範囲内で以上と同様の処理を行う。これにより、間引きデータD6から6クロックの範囲内で全ての誤差が所望の値以下となり、かつ、間引きデータD6からの時間間隔が最も長くなるサンプル点を次の標本点として検出する。

**【0183】**

以下同様にして、複数の標本点を順次検出していく。そして、検出した各標本点における離散的な振幅データ値と、各標本点間の時間間隔をクロックの数で表すタイミングデータ値との組を得る。上述の例では、各標本点における振幅データ値（D 1， D 6， …）とタイミングデータ値（5， ※， …）との組（D 1， 5）、（D 6， ※）…を得る。

#### 【 0 1 8 4 】

なお、ここでは、最初に 2 つの間引きデータ間の時間間隔が所定範囲内で最大となるサンプル点（間引きデータ D 1 と D 7 のサンプル点）を選んで誤差判定を開始し、時間間隔を順次短くしていく方向で処理を進めていく例について説明したが、標本点検出の方法はこれに限定されない。

#### 【 0 1 8 5 】

例えば、最初に 2 つの間引きデータ間の時間間隔が所定範囲内で最小となるサンプル点（間引きデータ D 1 と D 3 のサンプル点）を選んで誤差判定を開始し、時間間隔を順次長くしていく方向で処理を進めていっても良い。また、2 つの間引きデータ間の時間間隔が所定範囲内の中央付近となるサンプル点（例えば間引きデータ D 1 と D 4 のサンプル点）を選んで誤差判定を開始するようにしても良い。さらに、所定範囲内で採り得る時間間隔の全てのパターンについて誤差判定を行い、その後で、誤差の条件を満たしたパターンの中から時間間隔が最も長くなるパターンを選ぶようにしても良い。

#### 【 0 1 8 6 】

ところで、基準の間引きデータから所定範囲内において、誤差の条件を満たすサンプル点が 2 つ以上存在することもある。この場合、誤差の条件を満たす 2 つ以上のサンプル点の中から、基準の間引きデータからの時間間隔が最も長くなるサンプル点を標本点として検出するようにしている。このようにすることで、個々のタイミングデータの値を所定ビット内に収めることができるとともに、検出する標本点の数を極力減らすことができ、その分圧縮率を高めることができる。

#### 【 0 1 8 7 】

ここで、2 つの間引きデータ間を結ぶ直線上の各データ値と、その各データ値と同じサンプル点における間引きデータ値との誤差の演算方法を、図 1 6 に従っ

て説明する。図 1 6 (a) は、基準の間引きデータ D 1 と、そこから 2 クロック分離れた位置の間引きデータ D 3 とを直線で結んだ場合の誤差 e 2 を示す図である。

### 【 0 1 8 8 】

図 1 6 (a) において、データ D 1 - D 3 間にある間引きデータ D 2 と、データ D 1 - D 3 間を結ぶ直線上のデータ D 2' との誤差 e 2 は、以下の式(25)のように表せる。

$$\begin{aligned}
 e_2 &= (D_2 - D_1) - (D_3 - D_1) / 2 \\
 2e_2 &= 2D_2 - 2D_1 - D_3 + D_1 \\
 &= 2D_2 - D_1 - D_3 \\
 &= (D_2 - D_1) - (D_3 - D_2) \\
 &= D_2' - D_3' \\
 \therefore e_2 &= -(D_3' - D_2') / 2 = -D_3'' / 2 \quad \cdots (25)
 \end{aligned}$$

### 【 0 1 8 9 】

ただし、上記式(25)において、' の記号は 1 回微分値を示し、" の記号は 2 回微分値を示す。上記式(25)のように、直線の時間間隔が 2 クロックの場合の誤差 e 2 は、間引きデータ D 3 の二重微分値を用いて表すことができる。よって、許容誤差を  $\delta$  と置くと、図 1 4 の直線圧縮部 2 0 3 では、 $|e_2| \leq \delta$  の判定条件を満たすかどうかを判断することになる。ここで、| | の記号は絶対値を表す。

### 【 0 1 9 0 】

図 1 6 (b) は、基準の間引きデータ D 1 と、そこから 3 クロック分離れた位置の間引きデータ D 4 とを直線で結んだ場合の 2 つの誤差 e 2 , e 3 を示す図である。図 1 6 (b) において、データ D 1 - D 4 間にある間引きデータ D 2 , D 3 と、データ D 1 - D 4 間を結ぶ直線上のデータ D 2'' , D 3'' との誤差 e 2 , e 3 は、それぞれ以下の式(26)(27)のように表せる。

### 【 0 1 9 1 】

$$\begin{aligned}
 e_2 &= (D_2 - D_1) - (D_4 - D_1) / 3 \\
 3e_2 &= 3D_2 - 3D_1 - D_4 + D_1 \\
 &= 3D_2 - 2D_1 - D_4
 \end{aligned}$$

$$\begin{aligned}
&= 2(D_2 - D_1) - (D_4 - D_2) \\
&= 2(D_2 - D_1) - (D_4 - D_3) - (D_3 - D_2) \\
&= 2D_2' - D_4' - D_3' \\
&= -(D_4' - D_3') - 2(D_3' - D_2') \\
&= -D_4'' - 2D_3'' \\
\therefore e_2 &= -1/3(D_4'' + 2D_3'') \quad \cdots(26)
\end{aligned}$$

【 0 1 9 2 】

$$\begin{aligned}
e_3 &= (D_3 - D_1) - 2(D_4 - D_1) / 3 \\
3e_3 &= 3D_3 - 3D_1 - 2D_4 + 2D_1 \\
&= 3D_3 - D_1 - 2D_4 \\
&= -2(D_4 - D_3) + D_3 - D_1 \\
&= -2(D_4 - D_3) + (D_3 - D_2) + (D_2 - D_1) \\
&= -2D_4' + D_3' + D_2' \\
&= -2(D_4' - D_3') - (D_3' - D_2') \\
&= -2D_4'' - D_3'' \\
\therefore e_3 &= -1/3(2D_4'' + D_3'') \quad \cdots(27)
\end{aligned}$$

【 0 1 9 3 】

上記式(26)(27)のように、直線の時間間隔が3クロックの場合の誤差  $e_2$  ,  $e_3$  は、共にサンプリングデータ  $D_3$  ,  $D_4$  の二重微分値を用いて表すことができる。この場合、図14の直線圧縮部203では、 $|e_2| \leq \delta$ 、かつ、 $|e_3| \leq \delta$ の判定条件を満たすかどうかを判断することになる。なお、誤差  $e_2$  ,  $e_3$  のうちどちらが大きいかをまず判断し、大きい方だけを用いて判定条件を満たすか否かを判断するようにしても良い。

【 0 1 9 4 】

同様に、直線の時間間隔が4クロックの場合の誤差  $e_2$  ,  $e_3$  ,  $e_4$  は、間引きデータ  $D_3$  ,  $D_4$  ,  $D_5$  の二重微分値を用いて以下の式(28)～(30)のように表すことができる。

$$e_2 = -1/4(D_5'' + 2D_4'' + 3D_3'') \quad \cdots(28)$$

$$e_3 = -1/4(2D_5'' + D_4'' + 2D_3'') \quad \cdots(29)$$

$$e_4 = -1/4(3D_5'' + 2D_4'' + D_3'') \quad \cdots(30)$$

この場合、図 1 4 の直線圧縮部 2 0 3 では、 $|e_2| \leq \delta$ 、 $|e_3| \leq \delta$ 、かつ、 $|e_4| \leq \delta$  の判定条件を満たすかどうかを判断することになる。なお、誤差  $e_2$ 、 $e_3$ 、 $e_4$  のうちどれが一番大きいかをまず判断し、最も大きいものだけを用いて判定条件を満たすか否かを判断するようにしても良い。

#### 【0 1 9 5】

同様に、直線の時間間隔が 5 クロックの場合の誤差  $e_2$ 、 $e_3$ 、 $e_4$ 、 $e_5$  は、サンプリングデータ  $D_3$ 、 $D_4$ 、 $D_5$ 、 $D_6$  の二重微分値を用いて以下の式(31)～(34)のように表すことができる。

$$-e_2 = 1/5(D_6'' + 2D_5'' + 3D_4'' + 4D_3'') \quad \cdots(31)$$

$$-e_3 = 1/5(2D_6'' + 4D_5'' + 6D_4'' + 3D_3'') \quad \cdots(32)$$

$$-e_4 = 1/5(3D_6'' + 6D_5'' + 4D_4'' + 2D_3'') \quad \cdots(33)$$

$$-e_5 = 1/5(4D_6'' + 3D_5'' + 2D_4'' + D_3'') \quad \cdots(34)$$

この場合、図 1 4 の直線圧縮部 2 0 3 では、 $|e_2| \leq \delta$ 、 $|e_3| \leq \delta$ 、 $|e_4| \leq \delta$ 、かつ、 $|e_5| \leq \delta$  の判定条件を満たすかどうかを判断することになる。なお、誤差  $e_2$ 、 $e_3$ 、 $e_4$ 、 $e_5$  のうちどれが一番大きいかをまず判断し、最も大きいものだけを用いて判定条件を満たすか否かを判断するようにしても良い。

#### 【0 1 9 6】

同様に、直線の時間間隔が 6 クロックの場合の誤差  $e_2$ 、 $e_3$ 、 $e_4$ 、 $e_5$ 、 $e_6$  は、サンプリングデータ  $D_3$ 、 $D_4$ 、 $D_5$ 、 $D_6$ 、 $D_7$  の二重微分値を用いて以下の式(35)～(39)のように表すことができる。

$$e_2 = -1/6(D_7'' + 2D_6'' + 3D_5'' + 4D_4'' + 5D_3'') \quad \cdots(35)$$

$$e_3 = -1/6(2D_7'' + 4D_6'' + 6D_5'' + 8D_4'' + 4D_3'') \quad \cdots(36)$$

$$e_4 = -1/6(3D_7'' + 6D_6'' + 9D_5'' + 6D_4'' + 3D_3'') \quad \cdots(37)$$

$$e_5 = -1/6(4D_7'' + 8D_6'' + 6D_5'' + 4D_4'' + 2D_3'') \quad \cdots(38)$$

$$e_6 = -1/6(5D_7'' + 4D_6'' + 3D_5'' + 2D_4'' + D_3'') \quad \cdots(39)$$

この場合、図 1 4 の直線圧縮部 2 0 3 では、 $|e_2| \leq \delta$ 、 $|e_3| \leq \delta$ 、 $|e_4| \leq \delta$ 、 $|e_5| \leq \delta$ 、かつ、 $|e_6| \leq \delta$  の判定条件を満たすかどうかを判

断することになる。なお、誤差  $e_2$  ,  $e_3$  ,  $e_4$  ,  $e_5$  ,  $e_6$  のうちどれが一番大きいかをまず判断し、最も大きいものだけを用いて判定条件を満たすか否かを判断するようにしても良い。

#### 【0 1 9 7】

上記式(25)～(39)に示したように、本実施形態の直線圧縮処理で用いる誤差データは全て、間引きデータの二重微分値だけを用いて求めることができる。そこで、直線圧縮部 2 0 3 は、間引きデータの二重微分値により誤差データを求め、誤差の条件を満たすサンプル点を標本点として検出する。そして、検出した標本点の振幅データと標本点間の時間間隔を表すタイミングデータとを出力する。

#### 【0 1 9 8】

次いで、丸め演算部 2 0 4 は、本発明の丸め演算手段に相当するものであり、直線圧縮部 2 0 3 より出力された振幅データの下位数ビットを丸める演算を行う。この丸め演算は、例えば、直線圧縮部 2 0 3 からの出力データを所定の値（例えば 2 5 6 または 5 1 2）で除算することにより行うことが可能である。このような丸め演算を行うことにより、1ワードにつきデータ長を数ビット削減することができ、ここでデータ量を大幅に削減することができる。

#### 【0 1 9 9】

丸め演算部 2 0 4 に入力される振幅データは、例えば符号付 1 6 ビットのデータであり、3 2 7 6 7 までの大きなデータ値を表現することが可能である。しかし、可聴音の音声データの場合、実際に使用されるデータは比較的大きな値であることが多く、1 6 ビットで表される全体のデータ領域（0 ～ 3 2 7 6 7）のうち、比較的小さなデータ領域にはデータは余り現れない。したがって、このように大きな値を有するデータに対して下位数ビットを削減しても、再生音声の品質に影響を与えることはほとんどない。

#### 【0 2 0 0】

また、丸め演算の処理として、上述のように単純に所定の値で除算するのではなく、丸め演算部 2 0 4 の入力データと出力データとの関係が非線形な関係となるようにデータ値を丸めるようにしても良い。図 1 7 は、この場合の丸め演算の処理例を示す図である。この図 1 7 の例では、直線圧縮部 2 0 3 より入力される

データ値の対数を取り、その値を丸め演算部 204 の出力データとしている。

#### 【0201】

また、図 18 に示すように、図 17 の対数曲線を幾つかの線分の結合により近似し、この近似関数を用いて、直線圧縮部 203 より入力されるデータの振幅に応じて、出力するデータの振幅を変えることにより、入力データ値と出力データ値とが非線形の関係となるようにしても良い。なお、この図 18 の例では 5 つの線分の結合により図 17 の対数曲線を近似しているが、これ以上の線分を結合させるようにすれば、精度をより向上させることができる。

#### 【0202】

このように非線形な関数に従って丸め演算を行うことにより、16 ビットで表される可聴音の音声データに関する全体のデータ領域のうち、ほとんどのデータが現れる中央付近のデータ領域に出力データ値を集中させることができる。これにより、丸め演算の影響を緩和することができるとともに、振幅の小さいところでの量子化ノイズを低減することができ、丸め演算を行うことによる再生音声の品質への影響を更に少なくすることができる。

#### 【0203】

なお、図 17 のように対数関数を用いて丸め演算を行う場合、対数の底の値によって再生音声の品質が多少変わってくる。そこで、この対数の底の値をパラメータとして任意に設定可能なようにしても良い。また、対数化されたデータ値の小数点以下の値をどのように扱うかによって、圧縮率および再生音声の品質に影響を及ぼす。そこで、圧縮率を重視するために小数点以下のデータ値を切り捨てるか、再生音声の品質を重視するために小数点以下のデータ値を何桁まで採用するのかなどをパラメータとして任意に設定できるようにしても良い。

#### 【0204】

なお、図 14 の例では、最初に標本点における振幅データを抽出し、抽出した振幅データに対して丸め演算を行うようにしているが、これとは逆に、最初に無音処理後の全データまたは間引き処理後の全間引きデータに対して丸め演算を行い、丸め演算が施された振幅データの中から標本点における振幅データを抽出するようにしても良い。

**【0205】**

次いで、図14のブロック化部205は、上記丸め演算部204によって下位数ビットの丸め演算が施された振幅データと、直線圧縮部203により生成されたタイミングデータとにヘッダ情報を付加して適当にブロック化し、圧縮データとして出力する。出力された圧縮データは、伝送路上に伝送され、または記録媒体に記録される。

**【0206】**

図19は、上記図14のように構成した圧縮装置に対する第3の実施形態による伸長装置の構成例を示す図である。図19に示すように、本実施形態の伸長装置は、逆ブロック化部211、逆丸め演算部212、直線伸長部213および補間処理部214を備えて構成されている。

**【0207】**

逆ブロック化部211は、ブロック化された圧縮データ中に含まれるヘッダ情報に基づいて、当該圧縮データ中から振幅データとタイミングデータとを取り出す。逆丸め演算部212は、本発明の逆丸め演算手段に相当するものであり、逆ブロック化部211により取り出された振幅データに対して、圧縮側の丸め演算部204とは反対の逆丸め演算を行う。

**【0208】**

すなわち、図14の丸め演算部204において、単純にデータ値を256または512などの所定の値で除算する処理を行った場合は、逆丸め演算部212では、振幅データを256倍または512倍する処理を行う。また、丸め演算部204において、図17、図18のような対数関数あるいはその近似関数を用いた非線形な丸め演算を行った場合は、これらの関数と逆の関係にある図20、図21のような指数関数あるいはその近似関数を用いて非線形な逆丸め演算を行う。

**【0209】**

なお、図20のような指数関数を用いて逆丸め演算を行う場合、指数の値をパラメータとして任意に設定可能なようにしても良い。また、指数化されたデータの小数点以下のデータ値を切り捨てるか、小数点以下のデータ値を何桁まで採用するのかなどもパラメータとして任意に設定できるようにしても良い。この場合



、圧縮時には小数点以下を切り捨て、伸長時には小数点以下のデータ値も採用することにより、圧縮率を高めつつ、再生音声の品質の向上を図ることができる。

#### 【0 2 1 0】

直線伸長部 2 1 3 は、本発明の第 1 の補間手段に相当するものであり、逆丸め演算部 2 1 2 より出力された振幅データと、逆ブロック化部 2 1 1 より出力されたタイミングデータとを用いて、連続する標本点の振幅データ間を直線的に補間する補間演算を順次行うことにより、個々の振幅データ値の間を補間する補間データを生成する。

#### 【0 2 1 1】

すなわち、直線伸長部 2 1 3 は、圧縮データ中に含まれるタイミングデータを逆ブロック化部 2 1 1 より入力して、圧縮側で検出された標本点間と同じ不定の時間間隔を表す読み出しクロックを入力クロック C K から生成する。そして、その生成した読み出しクロックに従って、逆丸め演算部 2 1 2 より逆丸め演算後の振幅データを 2 つずつ入力し、当該 2 つの振幅データ間を直線で補間する演算を行い、各標本点間の補間データを生成する。

#### 【0 2 1 2】

このように、直線伸長の処理は、逆丸め演算の施された各標本点における振幅データの間を、タイミングデータで示される時間間隔だけ直線補間するだけである。本実施形態の圧縮時には、2 つの間引きデータ間を直線補間した場合に、当該 2 つの間引きデータの間に他の間引きデータが、補間した直線とどれだけ誤差が生じるかを見て、直線補間をしても誤差が大きくなりえない点を標本点として検出するようにしている。したがって、このようにして得た各標本点の振幅データ間を単純に直線補間するだけでも、元の間引きデータとほぼ同じ波形のデータを再現することができる。

#### 【0 2 1 3】

補間処理部 2 1 4 は、本発明の第 2 の補間手段に相当するものであり、直線伸長部 2 1 3 より出力された離散的な第 1 の補間データに対し、第 1 の実施形態あるいは第 2 の実施形態で説明したような何れかの補間処理を行うことにより、第 2 の補間データを生成する。何れかの補間処理を選択的に切り替えて適用するこ

とができるようにしても良い。このようにして生成されたデジタル補間データは、必要に応じて図示しないD/A変換部によってアナログ信号に変換され、再生アナログ信号として出力される。

#### 【0 2 1 4】

上記のように構成した本実施形態による圧縮装置および伸長装置は、例えば、CPUあるいはMPU、ROM、RAMなどを備えたコンピュータシステムによって構成され、その機能は上述のROMやRAMなどに格納されたプログラムが動作することによって実現される。

また、上記のように構成した本実施形態による圧縮装置および伸長装置は、ロジック回路を組み合わせるハードウェア的に構成することも可能である。

#### 【0 2 1 5】

図22は、第3の実施形態を適用した場合の圧縮前の元データと伸長による再生データとを比較した波形図である。この図22に示すように、元データの波形と再生データの波形はほとんど一致しており、図面上ではあたかも1つの波形しか存在しないように見えている。

#### 【0 2 1 6】

図23は、圧縮前の元データと伸長による再生データとの入出力特性を示す図である。この図23からも分かるように、極めて良好な入出力特性が得られており、元データと再生データとはほとんど一致している。

#### 【0 2 1 7】

図24は、図19に示した伸長装置に、図14の圧縮装置で生成された単一の圧縮データを入力した場合に得られる伸長データの周波数特性を示す図である。この図24に示すように、極めて良好な周波数特性が得られている。これによれば、いわゆる折り返しノイズが出るところにちょうどノッチフィルタが入ったような形になっており、折り返しノイズの発生を効果的に抑制することができる。

#### 【0 2 1 8】

以上詳しく説明したように、第3の実施形態によれば、圧縮側の間引き処理部202において、全域において1回以上微分可能な有限台の標本化関数の元となるデジタル基本関数に基づいて間引きデータを求めることにより、ここで8倍の

圧縮率を達成することができる。

#### 【0219】

さらに、直線圧縮部203において、間引き処理部202で求めた間引きデータの中から、誤差の条件を満たす標本点のデータのみを取り出して、これとタイミングデータとの組だけを得るようにしている。また、丸め演算部204において、直線圧縮部203により取り出された振幅データの下位数ビットを丸める演算を行っているので、1ワードにつきデータ長を数ビット削減することができ、ここでデータ量を大幅に削減することができる。

#### 【0220】

以上のことから、本実施形態によれば、全体として非常に高い圧縮率（8～数100程度）を達成することができる。

#### 【0221】

しかも、伸長側の補間処理部214においては、圧縮側の間引き処理部202で用いたデジタル基本関数と同じ関数を用いて補間データを求めることにより、圧縮前の元データをほぼ忠実に再現することができる。また、直線伸長部213に対応する直線圧縮部203では、伸長処理の際に直線補間を行っても元データとの誤差が所望の値より大きくならないサンプル点を標本点として検出しているので、ここでも元データへの再現性を良好にすることができる。

#### 【0222】

特に、直線補間により生成される標本点間の補間データは、圧縮前の元データと比べてその振幅の誤差が小さいだけでなく、位相ずれも非常に小さく抑えることができる。圧縮対象のデータとして音声を用いた場合、位相ずれは音色に大きく影響してくるが、本実施形態ではこの位相ずれがほとんどないため、元データの音色を忠実に再現することができる。

#### 【0223】

また、圧縮側で丸め演算を行っているが、その対象とされるほとんどの振幅データは、全データ領域のうち中央付近のデータ領域に集中して現れ、端付近のデータ領域には余り現れないので、下位数ビットを削減しても、伸長側における再生データの品質に影響を与えないようにすることができる。

## 【 0 2 2 4 】

しかも、本実施形態では、この丸め演算を、丸め演算前のデータ値と丸め演算後のデータ値との関係が非線形な関係となるように行っている。これにより、圧縮対象のデータとして音声を用いた場合、可聴音の音声データを表す全体のデータ領域のうち、ほとんどのデータが現れる中央付近のデータ領域に出力データ値を集中させ、丸め演算の影響を緩和することができ、伸長側における再生音声の品質への影響を更に抑制することができる。

## 【 0 2 2 5 】

また、本実施形態によれば、圧縮対象となるデータを時間／周波数変換することなく、時間軸上でそのまま圧縮・伸長することができる。しかも、間引きやそれに対応する補間処理は極めて簡単な四則演算のみで実現することができ、直線圧縮やそれに対応する直線伸長の処理は、補間演算の中でも特に簡単な直線補間の演算を行うだけで実現することができる。

## 【 0 2 2 6 】

したがって、全体としての処理が複雑にならず、処理を高速化することができるとともに、構成を簡素化することもできる。圧縮側から圧縮データを伝送して伸長側で再生する場合には、時間軸上での非常に簡単な直線補間演算によって、入力される圧縮データを順次に処理して再生することができるので、リアルタイム動作を実現することができる。

## 【 0 2 2 7 】

以上のことから、本実施形態によれば、再生データの品質を極めて良好に保ったまま、より高い圧縮率を達成することができる。しかも、そのための演算時間を短縮化できるとともに、その演算回路を簡素化することができる。

## 【 0 2 2 8 】

なお、上記第 3 の実施形態では、伸長側において、各標本点における振幅データの間の補間データを直線補間によって求めているが、補間演算はこの例に限定されるものではない。例えば、所定の標本化関数を用いた曲線補間によって求めるようにしても良い。また、本出願人が先に出願した特願平 1 1 - 1 7 3 2 4 5 号等に記載した補間処理を行っても良い。この場合には、極めてアナログに近い

波形を補間そのもので得ることができるので、補間処理の後段のD/A変換器やLPFを不要とすることもできる。

#### 【0229】

また、伸長側だけでなく、圧縮側においても曲線補間を用い、圧縮対象のデータ中に含まれる2つの間引きデータ間で曲線補間を行ったときに元データとの誤差が所望の値以下となるサンプル点を標本点として順次検出するようにしても良い。この場合の曲線補間は、伸長側における曲線補間と同じ演算であることが好ましい。

#### 【0230】

また、上記第3の実施形態では、タイミングデータのビット数を3ビットとし、基準の間引きデータから6クロックの範囲内で直線を引いて誤差判定を行うようにしたが、本発明はこの例に限定されるものではない。例えば、誤差判定を行う際の所定範囲を7クロックとしても良い。また、タイミングデータのビット数を4ビット以上とし、基準の間引きデータから直線を引いて誤差判定を行う際の所定範囲を8クロック以上としても良い。このようにすれば、圧縮率を更に高めることが可能である。また、このタイミングデータのビット数、あるいは誤差判定を行う際の所定範囲をパラメータとして任意に設定できるようにしても良い。

#### 【0231】

また、上記第3の実施形態では、標本点を検出するための誤差判定を行う際に採用する2つの間引きデータ間の時間間隔として採り得る値は、連続的なものであった。例えば、最大16クロック（タイミングデータのビット数は5ビット）の範囲内で誤差判定を行う場合に、基準の間引きデータからの時間間隔として採り得るタイミングデータの値は、2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16の何れかであった。

#### 【0232】

これに対して、所定の間隔より広い範囲で、時間間隔として採り得るタイミングデータの値を不連続とするようにしても良い。例えば、最大16クロックの範囲内で誤差判定を行う場合に、基準の間引きデータからの時間間隔として採りとり得るタイミングデータの値を、2, 3, 4, 5, 6, 7, 8, 9, 10, 12

， 14， 16， 18， 20， 24 などとしても良い。

#### 【0233】

このようにした場合には、1つの標本点を検出するために行う誤差判定の数を多くすることなく、2つの間引きデータ間の時間間隔として採り得る最大幅を大きくすることができる。これにより、振幅の変動があまり大きくない部分、例えば無音部分では、判定時間の遅延を招くことなく、標本点間の間隔をできるだけ長くとして標本点の数を極力減らすことができ、圧縮率を更に高くすることができる。

#### 【0234】

また、2つの間引きデータ間の時間間隔は、所定の間隔より狭い範囲（時間間隔が“10”以下の範囲）では連続的であるため、この部分ではより精度良く誤差判定を行うことができる。通常、基準のサンプル点から誤差が所望の値を超えるサンプル点までの時間間隔は、上記所定の間隔より狭い範囲に多く現れる。この誤差の条件を満たすケースが多くなる範囲において、採り得る時間間隔を離散的にすると、標本点の数が逆に増えてしまうことが考えられる。

#### 【0235】

例えば、基準の間引きデータからの時間間隔として採り得るタイミングデータの値を、2， 4， 6， …とした場合に、本来であれば誤差の条件を満たす最大の時間間隔が5クロックであったとしても、この5クロックについては誤差判定を行っていないので、実際に採用される時間間隔は4クロックとなってしまう。このような状態が幾つも発生すると、その分標本点の検出数が増えることになる。しかし、上記所定の間隔より狭い範囲において採り得る時間間隔を連続的にしておけば、検出する標本点の数を極力減らすことが可能となる。

#### 【0236】

また、上記第3の実施形態では、標本点を検出するための誤差判定を、基準のサンプル点から所定のクロックの範囲内に限定して行うようにしたが、本発明はこの例に限定されるものではない。例えば、離散的な標本点を検出する際に選ぶ2つのデータ間の時間間隔に所定範囲内という制限を設けることなく処理を行う。そして、誤差が所望の値を超えるサンプル点の直前のサンプル点を標本点とし

で順次検出するようにしても良い。このようにした場合は、標本点間の間隔をできるだけ長くにとって、検出する標本点の数を更に少なくすることができ、より高い圧縮率を実現することができる。

#### 【0 2 3 7】

また、上記第3の実施形態で採用する誤差の許容値は、例えば2 5 6、3 8 4、5 1 2のような値を用いることができるが、この数値に限定されるものではない。誤差の許容値を小さくすれば再生アナログ信号の再現性を重視した圧縮・伸長を実現することができる。また、誤差の許容値を大きくすれば圧縮率を重視した圧縮・伸長を実現することができる。この誤差許容値をパラメータとして任意に設定できるようにしても良い。

#### 【0 2 3 8】

また、誤差許容値をデータ振幅の関数とし、例えば振幅の大きいところで誤差許容値を大きくし、振幅の小さいところで誤差許容値を小さくするようにしても良い。振幅の大きいところでは、誤差がある程度大きくなってもそれが目立つことがなく、音質に大きな影響を与えることはない。したがって、このように誤差許容値をデータ振幅の関数として動的に変えるようにすれば、誤差がある程度大きくなってもそれが目立つことがないところでは、標本点間の間隔をできるだけ長くにとって、検出する標本点の数を極力減らすことができるとともに、誤差が比較的目立ちやすいところでは、誤差が大きくなるようにすることができる。これにより、再生データの音質を極めて良好に保ちながら、圧縮率を更に高めることができる。

#### 【0 2 3 9】

また、誤差許容値を周波数の関数とし、例えば周波数の高いところで誤差許容値を大きくし、周波数の低いところで誤差許容値を小さくするようにしても良い。圧縮対象として一連に入力される信号で周波数の高い部分、つまり近接するサンプル点においてもサンプリングデータ値が比較的大きく変化するような部分では、誤差許容値が小さいと検出される標本点の数が多くなり、高い圧縮率を実現できなくなることがある。しかし、周波数の高い部分で動的に誤差許容値を大きくすることにより、再生データの音質を全体として良好に保ちながら、圧縮率を

更に高めることが可能である。

#### 【0 2 4 0】

もちろん、誤差許容値をデータ振幅および周波数の両方の関数として動的に変化させるようにしても良い。

#### 【0 2 4 1】

また、上記第 3 の実施形態では、丸め演算を非線形的に行う場合、丸め演算前後のデータ値が対数関数の関係となるようにしたが、一部のデータ領域を重視するような関係であれば、対数関数以外も適用することが可能である。

#### 【0 2 4 2】

また、図 1 4 に示す圧縮装置の構成において、無音処理部 2 0 1 は省略することが可能である。

また、丸め演算部 2 0 4 を省略して画像データの圧縮に適用することも可能である。

#### 【0 2 4 3】

(第 4 の実施形態)

次に、本発明の第 4 の実施形態について説明する。

図 2 5 は、第 4 の実施形態による圧縮装置の構成例を示す図である。なお、図 2 5 において、図 1 4 に示した符号と同一の符号を付したものは同一の機能を有するものであるので、ここでは重複する説明を省略する。

#### 【0 2 4 4】

図 2 5 に示すように、本実施形態の圧縮装置は、フォーマット変換部 3 0 1、無音処理部 2 0 1、間引き処理部 2 0 2、丸め演算部 2 0 4、ゼロ圧縮部 3 0 2 およびブロック化部 3 0 3 を備えて構成されている。

#### 【0 2 4 5】

フォーマット変換部 3 0 1 は、入力信号のフォーマットを本実施形態の圧縮装置に適した形に変換する処理を行う。例えば、入力された W A V (ウェーブ) フォーマットの信号を T X T (テキスト) フォーマットの信号に変換する。入力信号がアナログ信号の場合は、そのアナログ信号を A / D 変換した後にフォーマット変換を行う。



**【 0 2 4 6 】**

ゼロ圧縮部 3 0 2 は、本発明のゼロ圧縮手段に相当するものであり、丸め演算部 2 0 4 の出力データについて、ゼロデータが  $n$  クロック以上続いたときに、その連続する  $n$  個以上のゼロデータを、 $-0$  (H 8 0) と、連続するゼロデータのクロック数との組に置き換えて出力する。上記丸め演算部 2 0 4 で対数関数を用いて丸め演算を行ったときの出力値は 2 5 5 までであり、 $-0$  (H 8 0) はデータ値として存在しないので、これをゼロ圧縮のスタートマークとして使用することができる。

**【 0 2 4 7 】**

ゼロ圧縮状態より回復するには、所定のしきい値以上のデータが  $m$  クロック（例えば 2 クロック）以上続いたことを検知して、ゼロ圧縮状態より回復して通常の動作に戻るようにする。

**【 0 2 4 8 】**

ブロック化部 3 0 3 は、ゼロ圧縮部 3 0 2 によってゼロ圧縮の処理が施された振幅データにヘッダ情報を付加して適当にブロック化し、圧縮データとして出力する。出力された圧縮データは、伝送路上に伝送され、または記録媒体に記録される。

**【 0 2 4 9 】**

図 2 6 は、ブロック化した圧縮データのフォーマット例を示す図である。図 2 6 (a) は圧縮データ全体のフォーマットを示す。図 2 6 (a) に示すように、圧縮データは、6 4 バイトのヘッダ情報の次に、ゼロ圧縮部 3 0 2 より出力された圧縮情報が続いて形成される。圧縮情報は、9 ビットを一単位としたデータである。

**【 0 2 5 0 】**

図 2 6 (b) は、圧縮情報のフォーマットを示す。有音部（ゼロ圧縮部 3 0 2 により置き換えが行われていないデータ）は、丸め演算部 2 0 4 より出力された圧縮情報から成る。また、無音部（ゼロ圧縮部 3 0 2 により置き換えが行われたデータ）は、 $-0$  (サインビットにのみ “1” が立っているデータ) と、連続するゼロデータのクロック数との組から成る。

**【 0 2 5 1 】**

図 2 7 は、上記図 2 5 のように構成した圧縮装置に対する第 4 の実施形態による伸長装置の構成例を示す図である。なお、図 2 7 において、図 1 9 に示した符号と同一の符号を付したものは同一の機能を有するものであるので、ここでは重複する説明を省略する。

**【 0 2 5 2 】**

図 2 7 に示すように、本実施形態の伸長装置は、逆ブロック化部 3 1 1、逆丸め演算部 2 1 2、補間処理部 2 1 4 およびフォーマット変換部 3 1 2 を備えて構成されている。この場合の補間処理部 2 1 4 は、本発明の補間手段に相当する。

**【 0 2 5 3 】**

逆ブロック化部 3 1 1 は、ブロック化された図 2 6 のような圧縮データ中に含まれるヘッダ情報に基づいて、当該圧縮データ中から圧縮情報を取り出す。このとき、- 1 (H 8 0) のデータを検知したときは、その後に続いて記録されているクロック数分だけゼロデータを再生する。このように、逆ブロック化部 3 1 1 は、本発明のゼロ伸長手段を含んでいる。

**【 0 2 5 4 】**

フォーマット変換部 3 1 2 は、補間処理部 2 1 4 から出力された伸長データのフォーマットを圧縮前の元の形式に変換する処理を行う。例えば、補間処理部 2 1 4 の出力データを W A V (ウェーブ) フォーマットの信号に変換する。また、必要に応じて、フォーマット変換後の信号を D / A 変換して出力する。

**【 0 2 5 5 】**

上記のように構成した本実施形態による圧縮装置および伸長装置は、例えば、C P U あるいは M P U、R O M、R A M などを備えたコンピュータシステムによって構成され、その機能は上述の R O M や R A M などに格納されたプログラムが動作することによって実現される。

また、上記のように構成した本実施形態による圧縮装置および伸長装置の一部あるいは全部は、ロジック回路を組み合わせるハードウェア的に構成することも可能である。

**【 0 2 5 6 】**

以上詳しく説明したように、第 4 の実施形態によれば、圧縮側の間引き処理部 2 0 2 において、全域において 1 回以上微分可能な有限台の標本化関数の元となるデジタル基本関数に基づいて間引きデータを求めることにより、ここで 8 倍の圧縮率を達成することができる。

#### 【 0 2 5 7 】

さらに、丸め演算部 2 0 4 において、間引き処理部 2 0 2 により生成された間引きデータ（振幅データ）の下位数ビットを丸める演算を行っているので、1 ワードにつきデータ長を数ビット削減することができ、ここでデータ量を大幅に削減することができる。

#### 【 0 2 5 8 】

第 3 の実施形態と比べると、直線圧縮による振幅データの削減はないが、その代わりタイミングデータは全く不要となる。したがって、圧縮情報を構成する一単位のデータ長を 9 ビットと短くすることができ、その分データ量を削減することができる。

#### 【 0 2 5 9 】

また、直線圧縮の代わりにゼロ圧縮部 3 0 2 においてゼロ圧縮の処理を行うことにより、ある時間間隔を有するゼロデータを“0”とクロック数との組に圧縮することが可能となり、データ量をより削減することができる。しかも、無音処理部 2 0 1 にて絶対値が小さなデータをあらかじめゼロデータに置き換えているので、このゼロ圧縮によるデータ削減量をより大きくすることができる。

#### 【 0 2 6 0 】

以上のことから、本実施形態によれば、全体として非常に高い圧縮率（8 ～ 数 1 0 0 程度）を達成することができる。

#### 【 0 2 6 1 】

しかも、伸長側の補間処理部 2 1 4 においては、圧縮側の間引き処理部 2 0 2 で用いたデジタル基本関数と同じ関数を用いて補間データを求めることにより、圧縮前の元データをほぼ忠実に再現することができる。

#### 【 0 2 6 2 】

また、圧縮側で丸め演算を行っているが、その対象とされるほとんどの振幅デ

ータは、全データ領域のうち中央付近のデータ領域に集中して現れ、端付近のデータ領域には余り現れないので、下位数ビットを削減しても、伸長側における再生データの品質に影響を与えないようにすることができる。

#### 【0 2 6 3】

しかも、本実施形態では、この丸め演算を、丸め演算前のデータ値と丸め演算後のデータ値との関係が非線形な関係となるように行っている。これにより、圧縮対象のデータとして音声を用いた場合、可聴音の音声データを表す全体のデータ領域のうち、ほとんどのデータが現れる中央付近のデータ領域に出力データ値を集中させ、丸め演算の影響を緩和することができ、伸長側における再生音声の品質への影響を更に抑制することができる。

#### 【0 2 6 4】

また、本実施形態によれば、圧縮対象となるデータを時間／周波数変換することなく、時間軸上でそのまま圧縮・伸長することができ、しかも、極めて簡単な四則演算のみで実現することができる。さらに、第3の実施形態と比べると、直線圧縮による処理も省略できる。したがって、全体としての処理が複雑にならず、構成を簡素化することもできる。また、圧縮側から圧縮データを伝送して伸長側で再生する場合には、時間軸上で非常に簡単な直線補間演算によって、入力される圧縮データを順次に処理して再生することができるので、リアルタイム動作を実現することができる。

#### 【0 2 6 5】

以上のことから、本実施形態によれば、再生データの品質を極めて良好に保ったまま、より高い圧縮率を達成することができる。しかも、そのための演算時間を短縮化できるとともに、その演算回路を簡素化することができる。

#### 【0 2 6 6】

以上に説明した第1～第4の実施形態による圧縮・伸長の手法は、上述のように、ハードウェア構成、DSP、ソフトウェアの何れによっても実現することが可能である。例えばソフトウェアによって実現する場合、本実施形態の圧縮装置および伸長装置は、実際にはコンピュータのCPUあるいはMPU、RAM、ROMなどで構成され、RAMやROMに記憶されたプログラムが動作することに

よって実現できる。

#### 【0 2 6 7】

したがって、コンピュータが上記本実施形態の機能を果たすように動作させるプログラムを例えば C D - R O M のような記録媒体に記録し、コンピュータに読み込ませることによって実現できるものである。上記プログラムを記録する記録媒体としては、C D - R O M 以外に、フロッピーディスク、ハードディスク、磁気テープ、光ディスク、光磁気ディスク、DVD、不揮発性メモリカード等を用いることができる。また、上記プログラムをインターネット等のネットワークを介してコンピュータにダウンロードすることによっても実現できる。

#### 【0 2 6 8】

また、コンピュータが供給されたプログラムを実行することにより上述の実施形態の機能が実現されるだけでなく、そのプログラムがコンピュータにおいて稼働している O S （オペレーティングシステム）あるいは他のアプリケーションソフト等と共同して上述の実施形態の機能が実現される場合や、供給されたプログラムの処理の全てあるいは一部がコンピュータの機能拡張ボードや機能拡張ユニットにより行われて上述の実施形態の機能が実現される場合も、かかるプログラムは本発明の実施形態に含まれる。

#### 【0 2 6 9】

なお、上記各実施形態は、本発明を実施するにあたっての具体化の一例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその精神、またはその主要な特徴から逸脱することなく、様々な形で実施することができる。

#### 【0 2 7 0】

##### 【発明の効果】

以上説明したように本発明によれば、圧縮率の向上と再生データの品質向上との両方を実現することができる。しかも、信号の圧縮・伸長処理を簡素化して処理時間を短くすることができるとともに、これを実現するための構成も簡単化することができる。

##### 【図面の簡単な説明】

**【図 1】**

第 1 の実施形態による圧縮装置（間引き装置）の構成例を示す図である。

**【図 2】**

第 1 の実施形態による伸長装置（補間装置）の構成例を示す図である。

**【図 3】**

第 1 の実施形態による間引きおよび補間の原理を説明するための図である。

**【図 4】**

第 1 ～ 第 4 の実施形態で用いるデジタル基本関数を示す図である。

**【図 5】**

図 2 に示した伸長装置にユニットパルスのデータを入力した場合に得られるオーバーサンプル結果を示す特性図である。

**【図 6】**

s i n c 関数の説明図である。

**【図 7】**

図 5 ④に示した標本化関数の周波数特性を示す図である。

**【図 8】**

第 1 の実施形態による伸長装置の他の構成例を示す図である。

**【図 9】**

第 1 の実施形態による伸長装置の他の構成例を示す図である。

**【図 1 0】**

第 1 の実施形態による伸長装置の他の構成例を示す図である。

**【図 1 1】**

第 2 の実施形態による間引きおよび補間の原理を説明するための図である。

**【図 1 2】**

第 2 の実施形態による圧縮装置（間引き装置）の構成例を示す図である。

**【図 1 3】**

第 2 の実施形態による伸長装置（補間装置）の構成例を示す図である。

**【図 1 4】**

第 3 の実施形態による圧縮装置の構成例を示す図である。

**【図 1 5】**

第 3 の実施形態による直線圧縮の基本原理を説明するための図である。

**【図 1 6】**

第 3 の実施形態による直線圧縮で用いる誤差演算を説明するための図である。

**【図 1 7】**

第 3 の実施形態による非線形的な丸め演算処理の一例を説明するための図である。

**【図 1 8】**

第 3 の実施形態による非線形的な丸め演算処理の他の例を説明するための図である。

**【図 1 9】**

第 3 の実施形態による伸長装置の構成例を示す図である。

**【図 2 0】**

第 3 の実施形態による非線形的な逆丸め演算処理の一例を説明するための図である。

**【図 2 1】**

第 3 の実施形態による非線形的な逆丸め演算処理の他の例を説明するための図である。

**【図 2 2】**

第 3 の実施形態を適用した場合の圧縮前の元データと伸長による再生データとを比較した波形図である。

**【図 2 3】**

第 3 の実施形態を適用した場合の圧縮前の元データと伸長による再生データとの入出力特性を示す図である。

**【図 2 4】**

図 1 9 に示した伸長装置に、図 1 4 の圧縮装置で生成された単一の圧縮データを入力した場合に得られる伸長データの周波数特性を示す図である。

**【図 2 5】**

第 4 の実施形態による圧縮装置の構成例を示す図である。

## 【図 2 6】

第 4 の実施形態においてブロック化された圧縮データのフォーマットの例を示す図である。

## 【図 2 7】

第 4 の実施形態による伸長装置の構成例を示す図である。

## 【符号の説明】

1<sub>-1</sub>, 2<sub>-1</sub>, 3<sub>-1</sub>, 4<sub>-1</sub> D型フリップフロップ

5<sub>-1</sub> 加算器

6<sub>-1</sub> 9 倍乗算器

7<sub>-1</sub> 加算器

8<sub>-1</sub> -1 倍乗算器

9<sub>-1</sub> 加算器

10<sub>-1</sub> 1/16 倍乗算器

11 D型フリップフロップ

21<sub>-1</sub>, 22<sub>-1</sub>, 23<sub>-1</sub> D型フリップフロップ

24<sub>-1</sub> -1 倍乗算器

25<sub>-1</sub>, 26<sub>-1</sub> ANDゲート

27<sub>-1</sub> インバータ

28<sub>-1</sub> ORゲート

29<sub>-1</sub> 8 倍乗算器

30<sub>-1</sub> -1 倍乗算器

31<sub>-1</sub>, 32<sub>-1</sub> ANDゲート

33<sub>-1</sub> インバータ

34<sub>-1</sub> ORゲート

35<sub>-1</sub>, 36<sub>-1</sub> 加算器

37<sub>-1</sub> D型フリップフロップ

38<sub>-1</sub> 加算器

39<sub>-1</sub> 1/16 倍乗算器

201 無音処理部

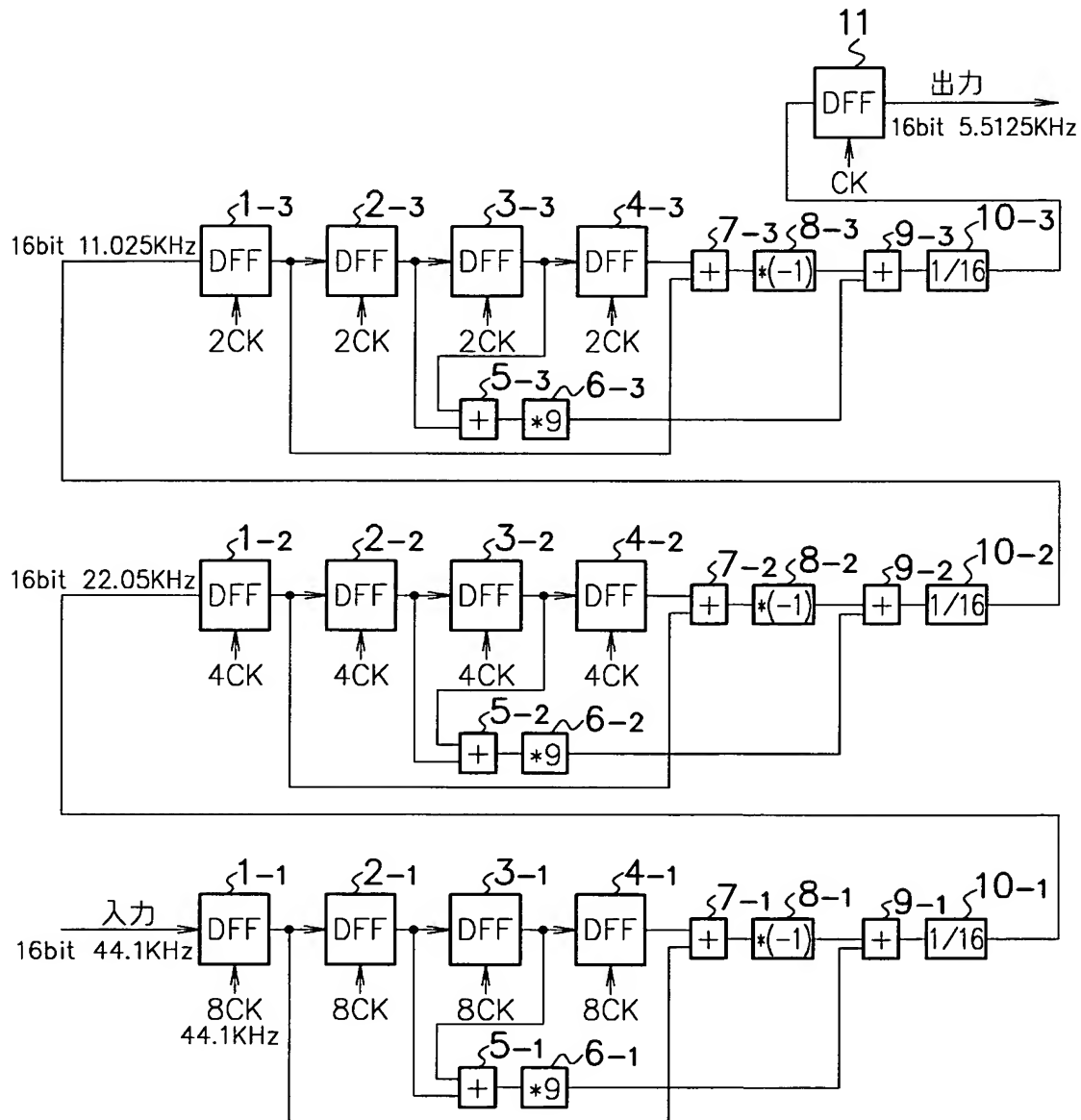


- 2 0 2 間引き処理部
- 2 0 3 直線圧縮部
- 2 0 4 丸め演算部
- 2 0 5 ブロック化部
- 2 1 1 逆ブロック化部
- 2 1 2 逆丸め演算部
- 2 1 3 直線伸長部
- 2 1 4 補間処理部
- 3 0 1 フォーマット変換部
- 3 0 2 ゼロ圧縮部
- 3 0 3 ブロック化部
- 3 1 1 逆ブロック化部
- 3 1 2 フォーマット変換部

【書類名】 図面

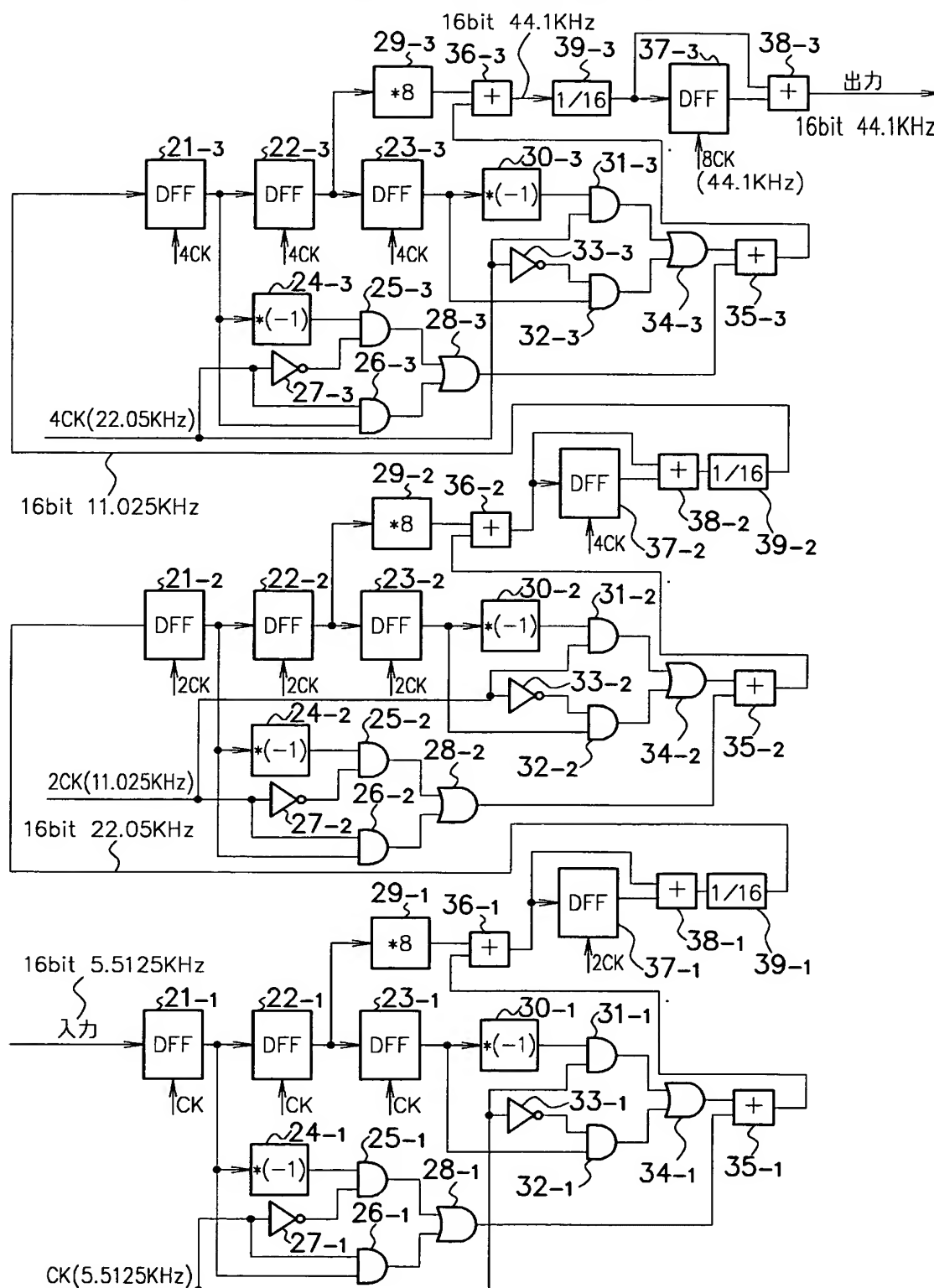
【図 1】

## 第1の実施形態による圧縮装置（間引き装置）



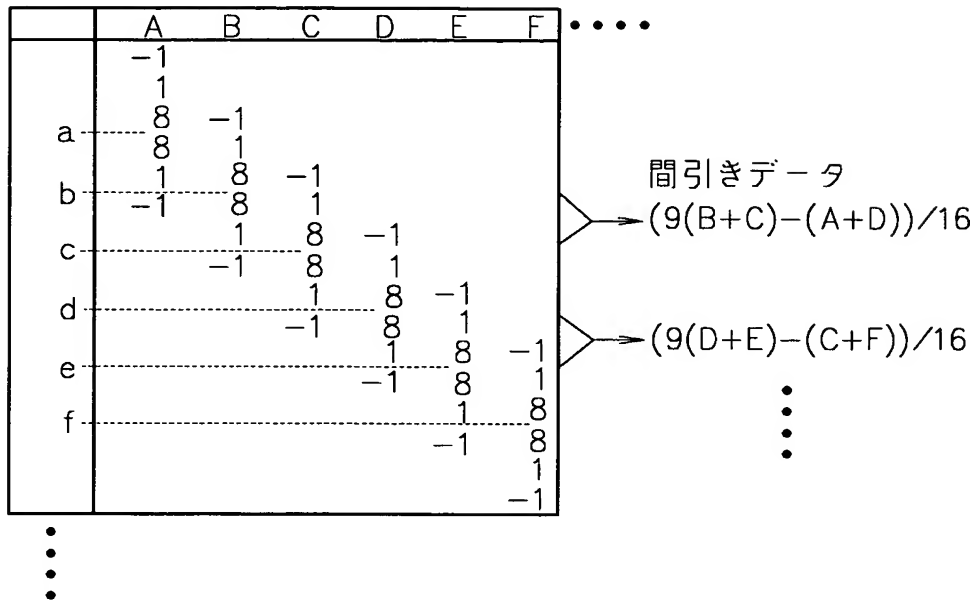
【図 2】

## 第1の実施形態による伸長装置（補間装置）

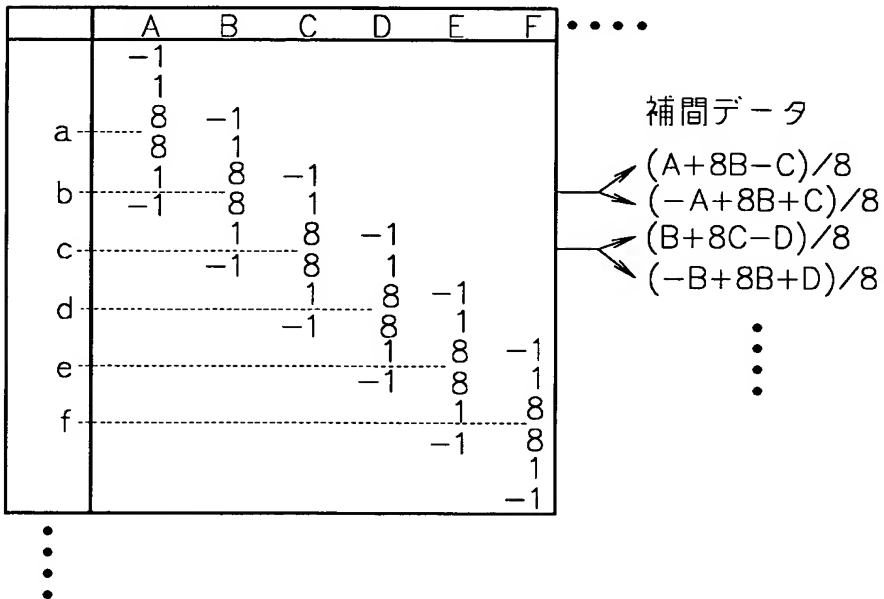


【図 3】

(a) 第 1 の実施形態の間引き原理

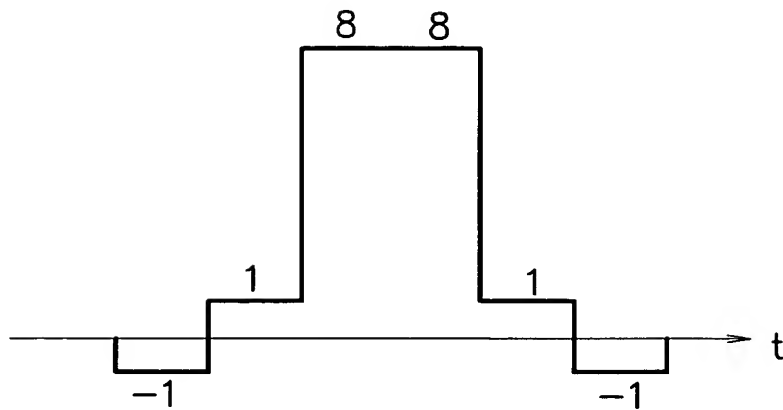


(b) 第 1 の実施形態の補間原理



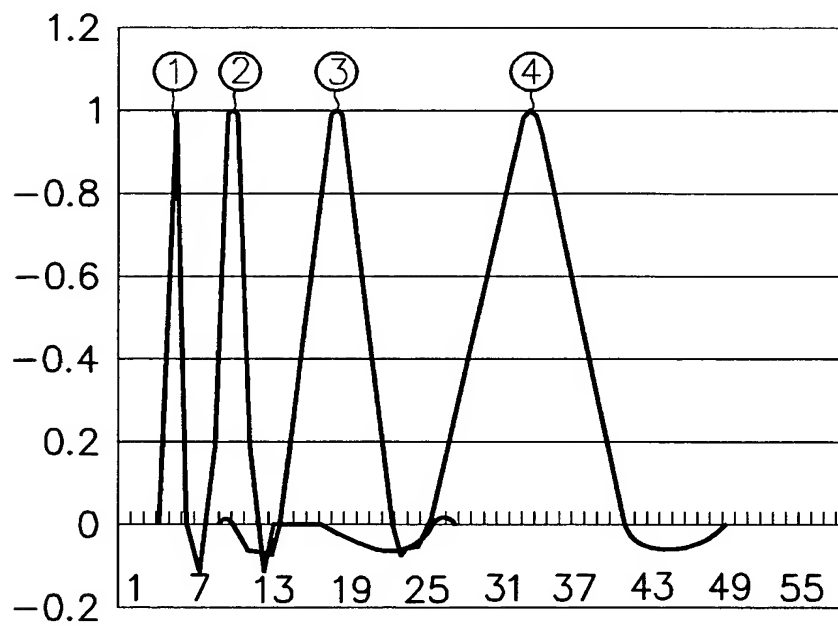
【図 4】

デジタル基本関数

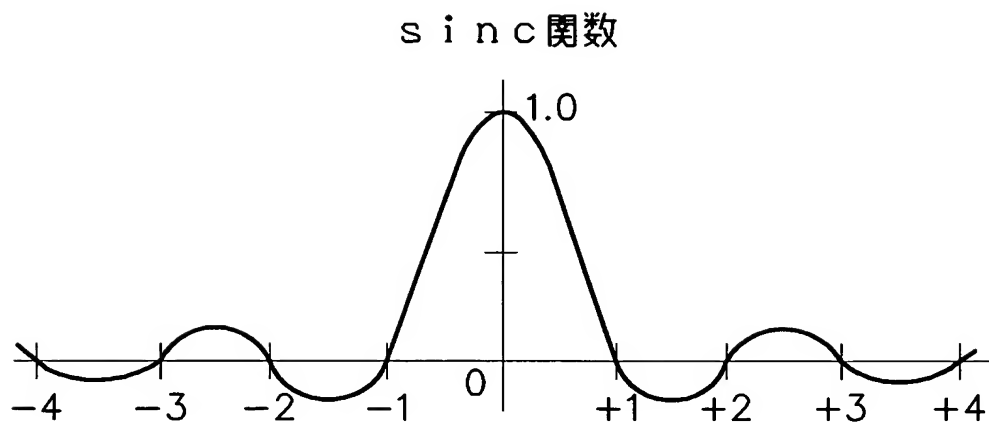


【図 5】

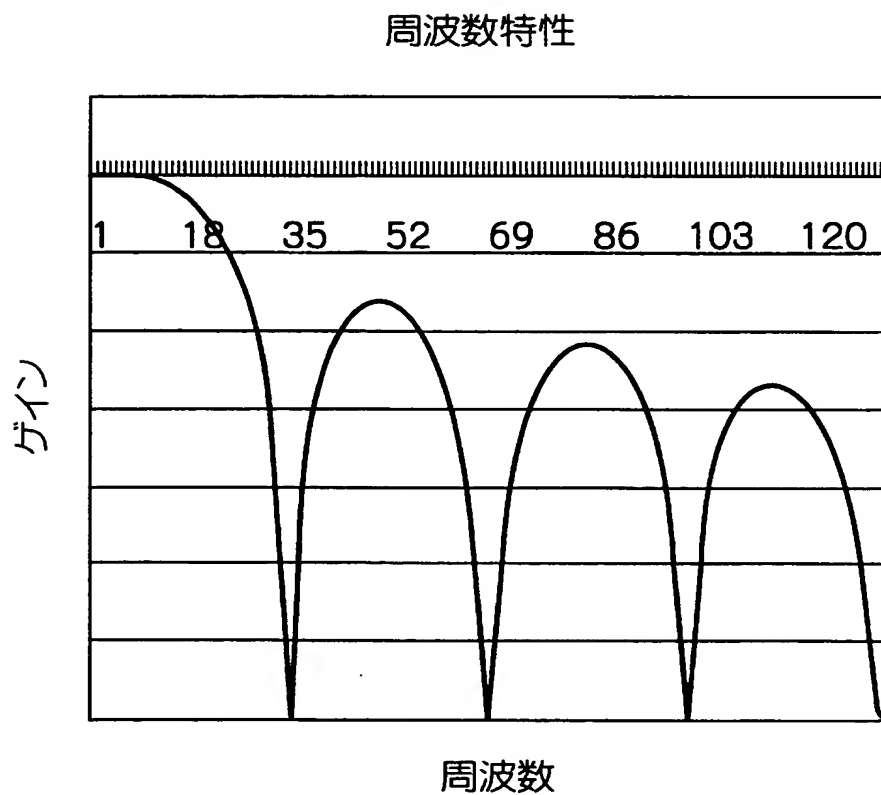
各段階のオーバーサンプル結果



【図 6】

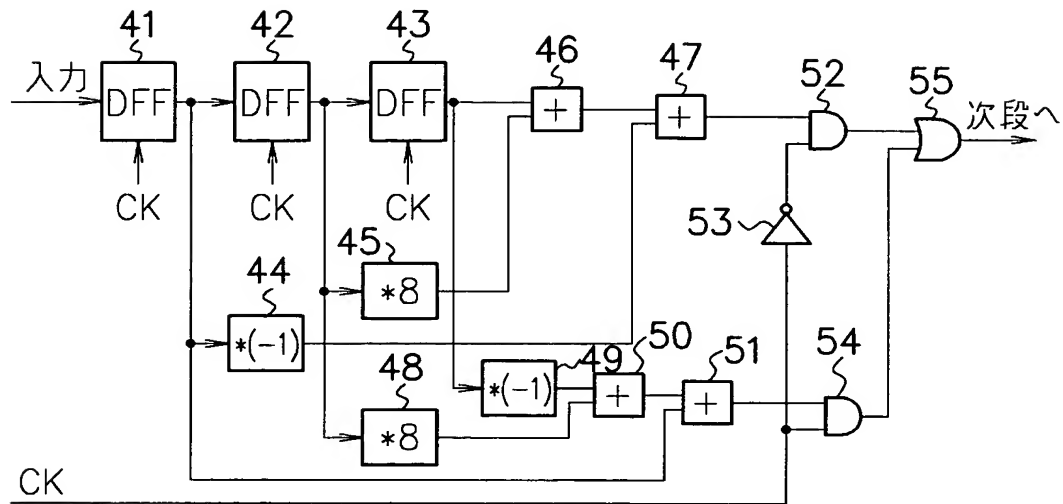


【図 7】



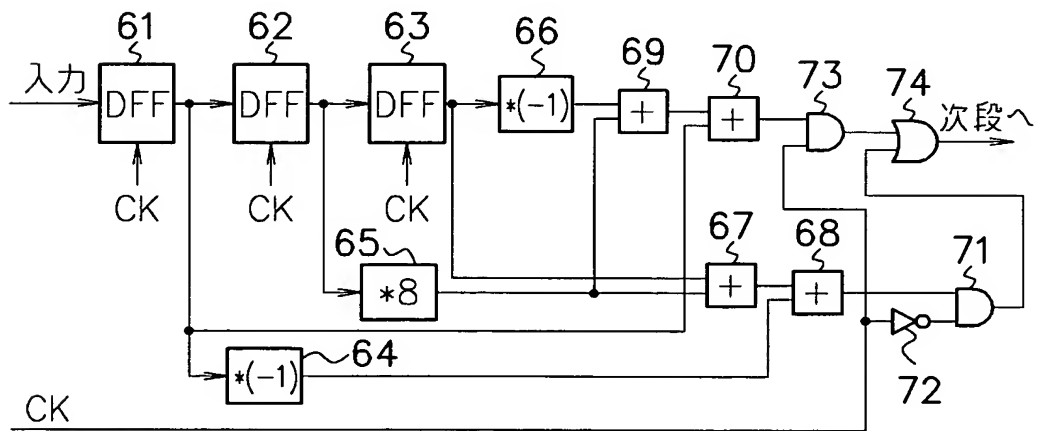
【図 8】

伸長装置の他の構成例



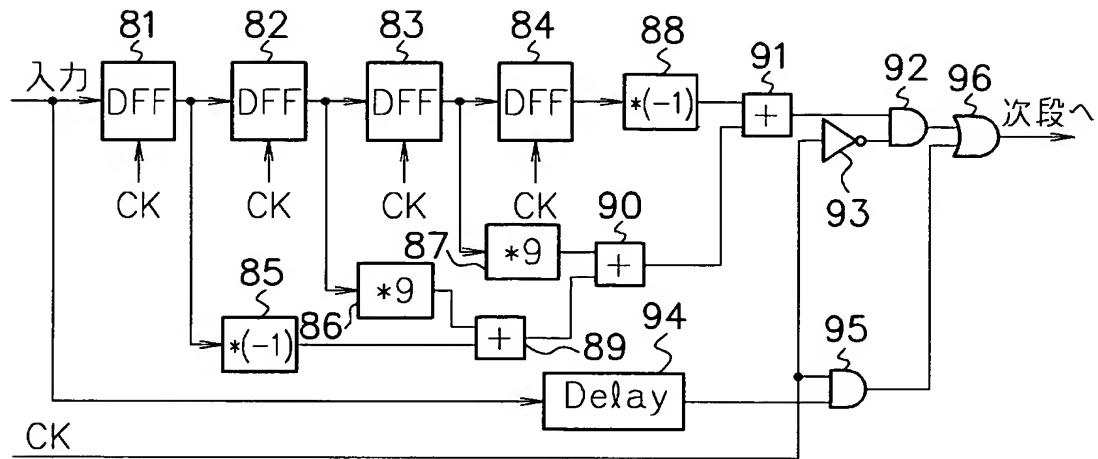
【図 9】

伸長装置の他の構成例



【図 1 0】

## 伸長装置の他の構成例





【図 11】

## (a) 第2の実施形態の間引き原理

	A	B	C	D	E	F	G	H	I	.....
	-1									
	0	-1								
	9	0	-1							
a	-16	9	0	-1						
b	9	-16	9	0	-1					
c	0	9	-16	9	0	-1				
d	-1	0	9	-16	9	0	-1			
e		-1	0	9	-16	9	0	-1		
f			-1	0	9	-16	9	0	-1	
g				-1	0	9	-16	9	0	
h					-1	0	9	-16	9	
i						-1	0	9	-16	
							-1	0	9	
								-1	0	
									-1	

間引きデータ

→  $(16D+9(C+E)-(A+G))/32$

→  $(16F+9(E+G)-(C+I))/32$

⋮

## (b) 第2の実施形態の補間原理

	A	B	C	D	E	F	.....
	9	-1					
a	-16	0					
	9	9	-1				
b	0	-16	0				
	-1	9	9	-1			
c	0	-16	0				
		-1	9	9	-1		
d			0	-16	0		
			-1	9	9	-1	
e				0	-16	0	
				-1	9	9	
f					0	-16	
					-1	9	
						0	
						-1	

補間データ

→  $(-A+9B+25C-D)/32$

→  $(-B+25C+9D-E)/32$

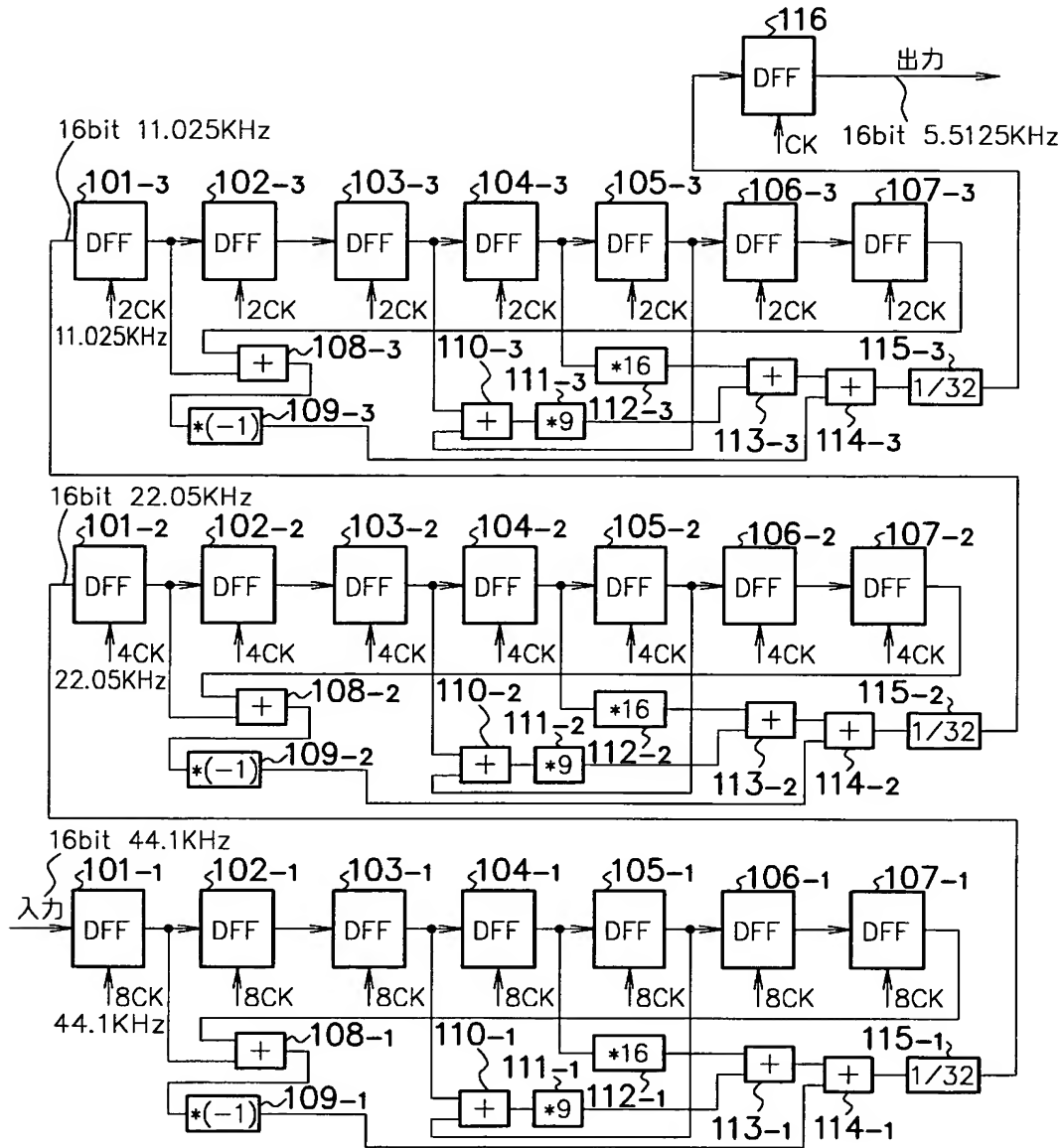
→  $(-B+9C+25D-E)/32$

→  $(-C+25D+9E-F)/32$

⋮

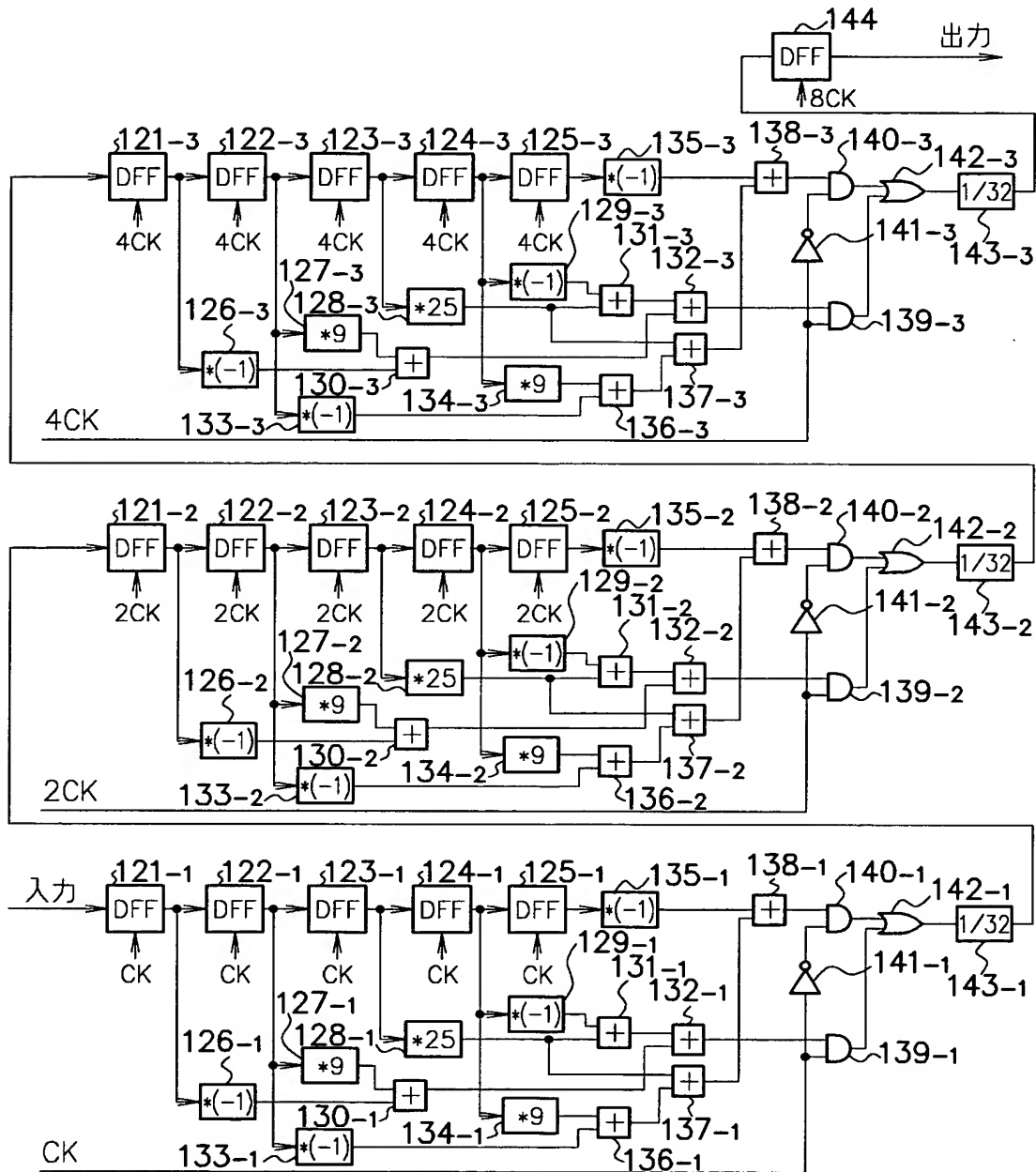
【図 12】

## 第2の実施形態による圧縮装置（間引き装置）



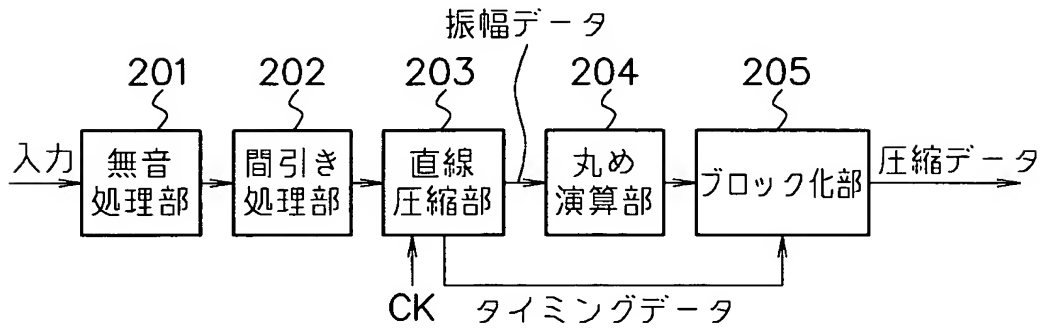
【図 13】

## 第2の実施形態による伸長装置（補間装置）



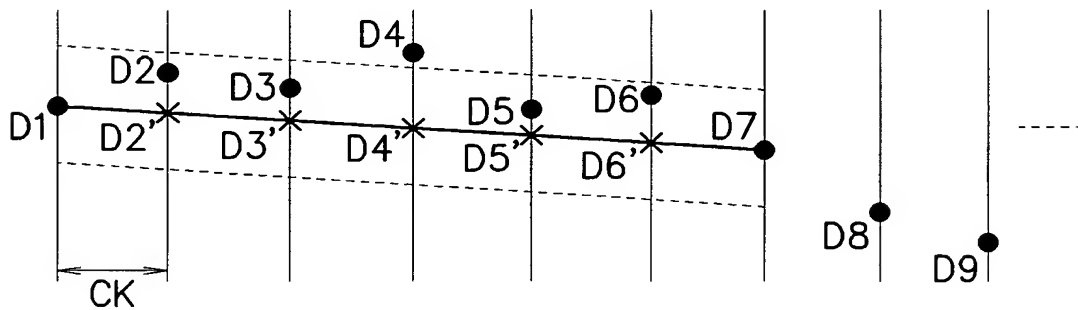
【図 14】

## 第3の実施形態による圧縮装置

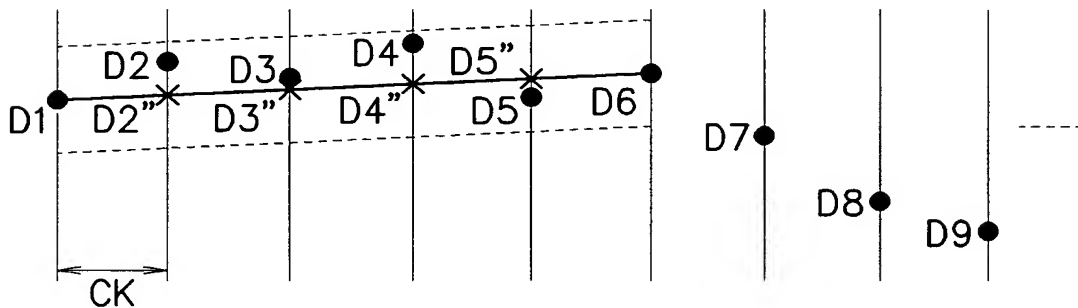


【図 15】

## 直線圧縮の原理



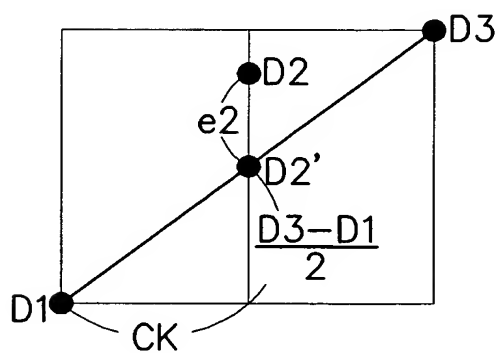
(a) D1 - D7間



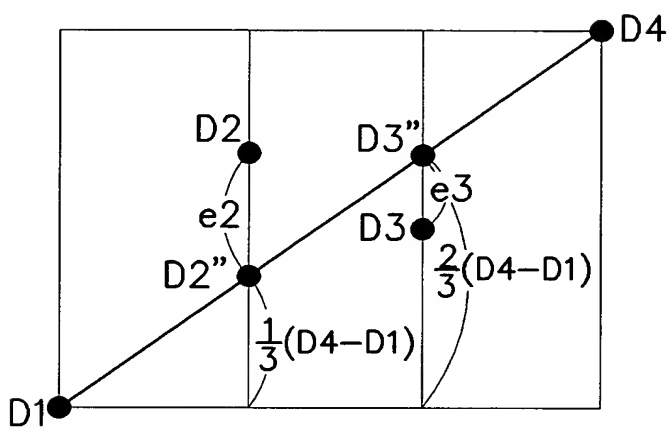
(b) D1 - D6間

【図 16】

誤差演算の説明図



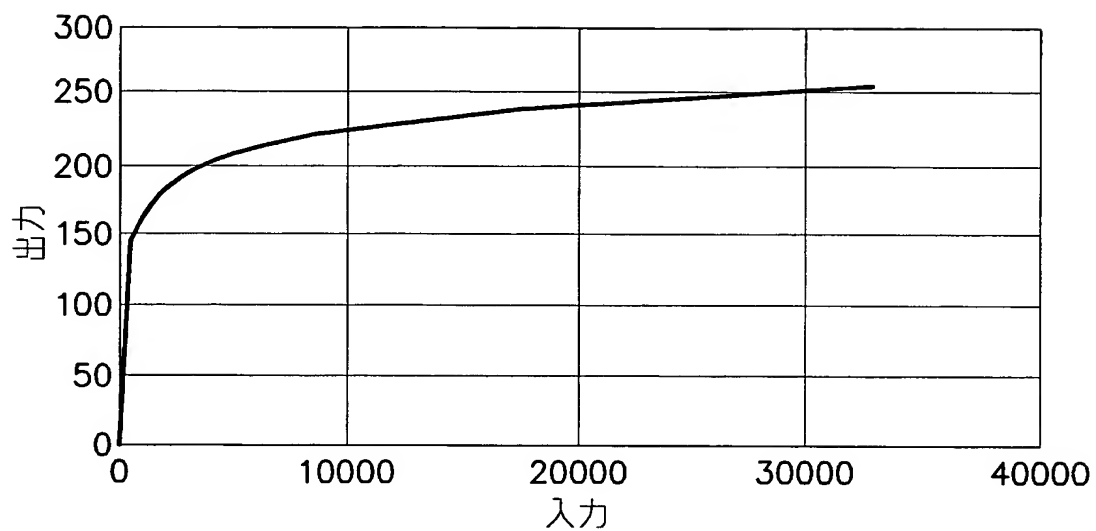
(a)



(b)

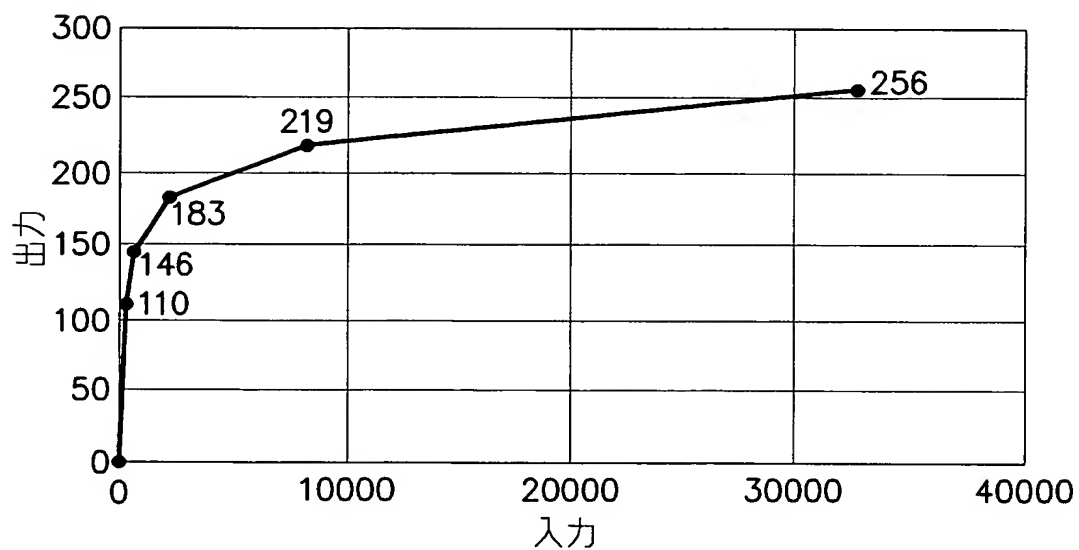
【図 1 7】

丸め演算処理の一例



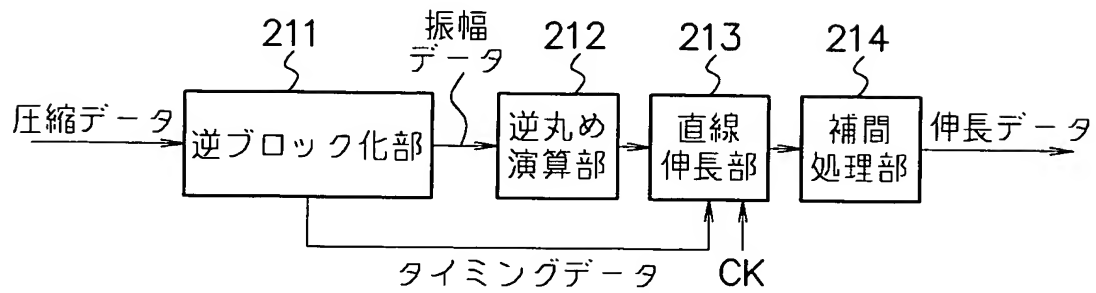
【図 1 8】

丸め演算処理の他の例



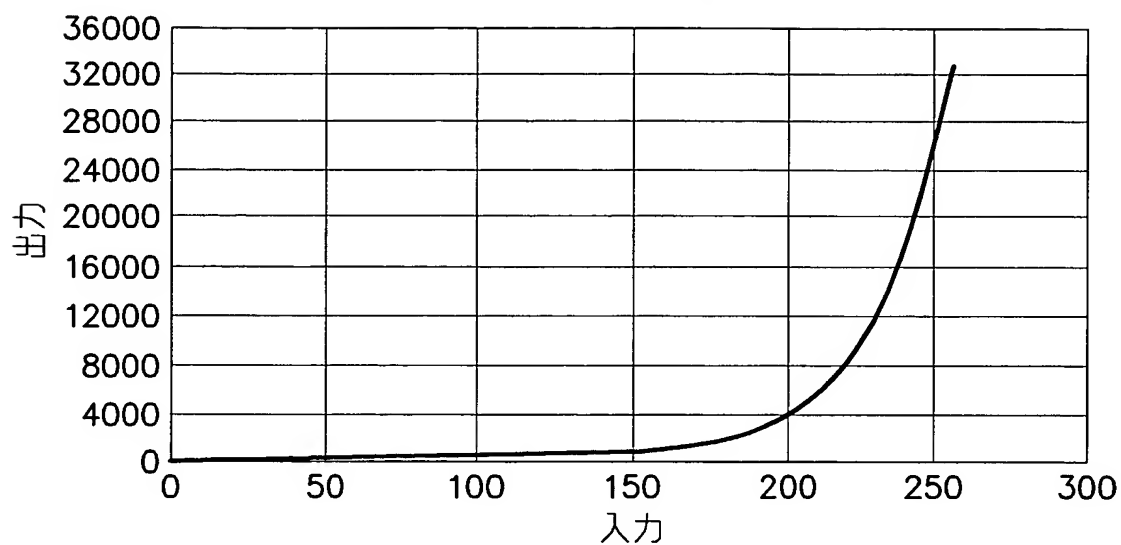
【図 19】

## 第3の実施形態による伸長装置



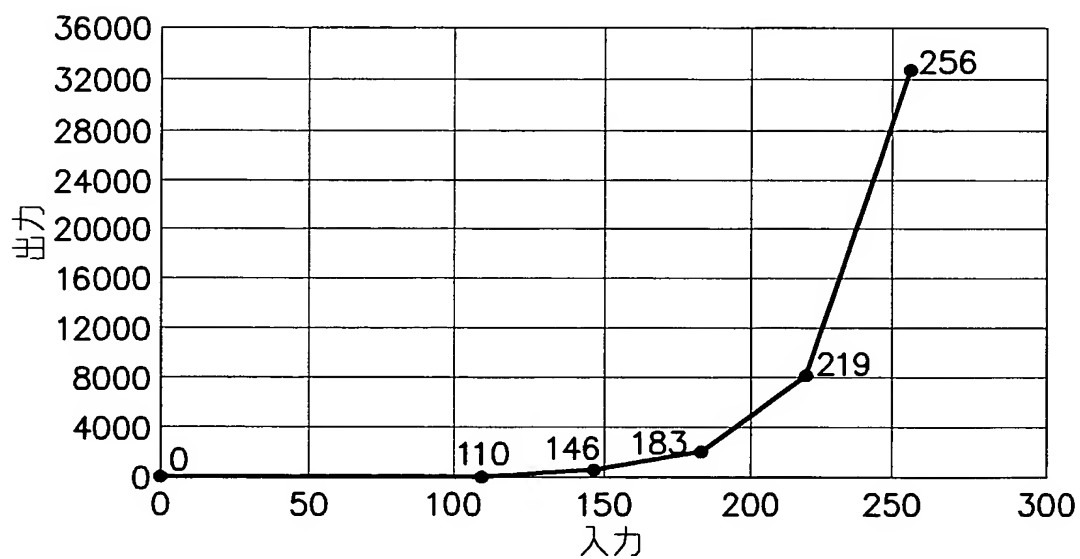
【図 2 0】

逆丸め演算処理の一例



【図 2 1】

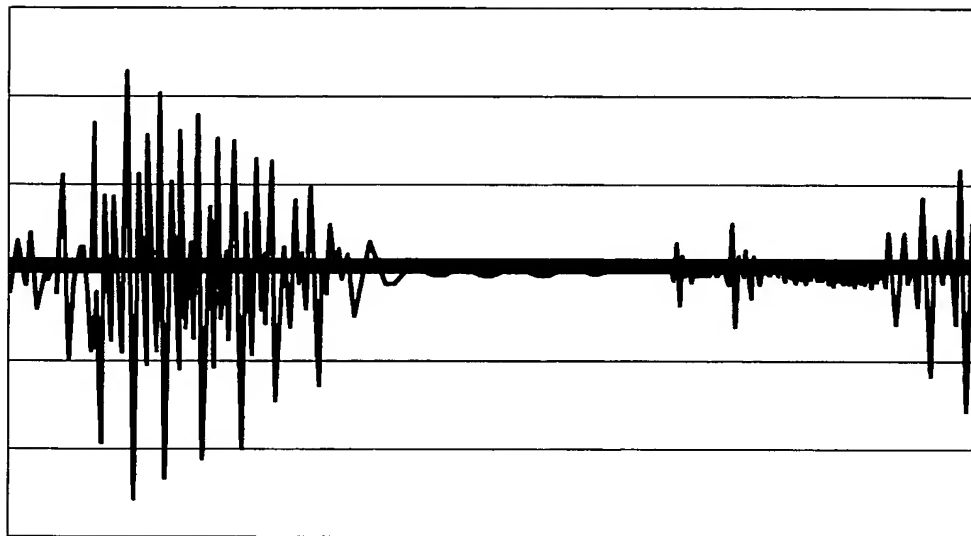
逆丸め演算処理の他の例





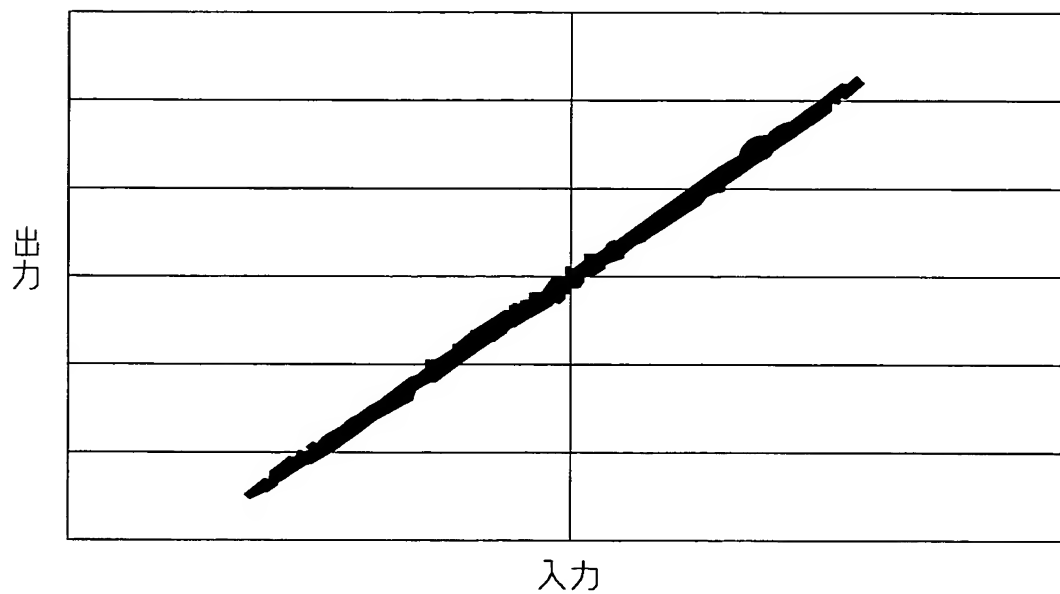
【図 2 2】

圧縮前の元データと伸長による再生データの波形

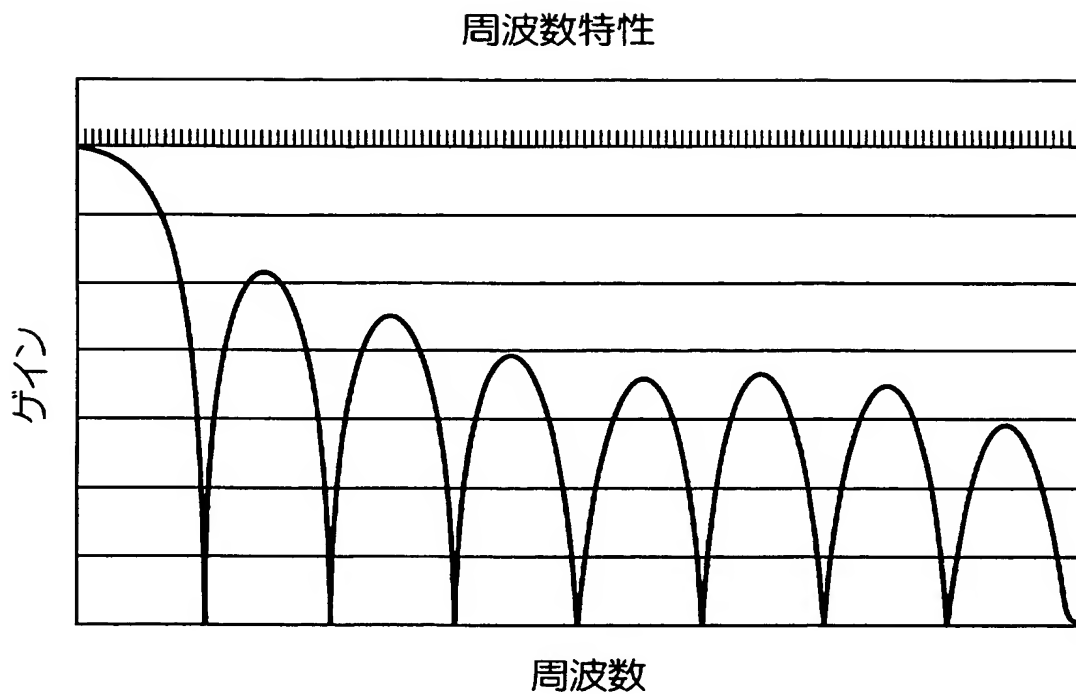


【図 2 3】

元データと再生データの入出力特性

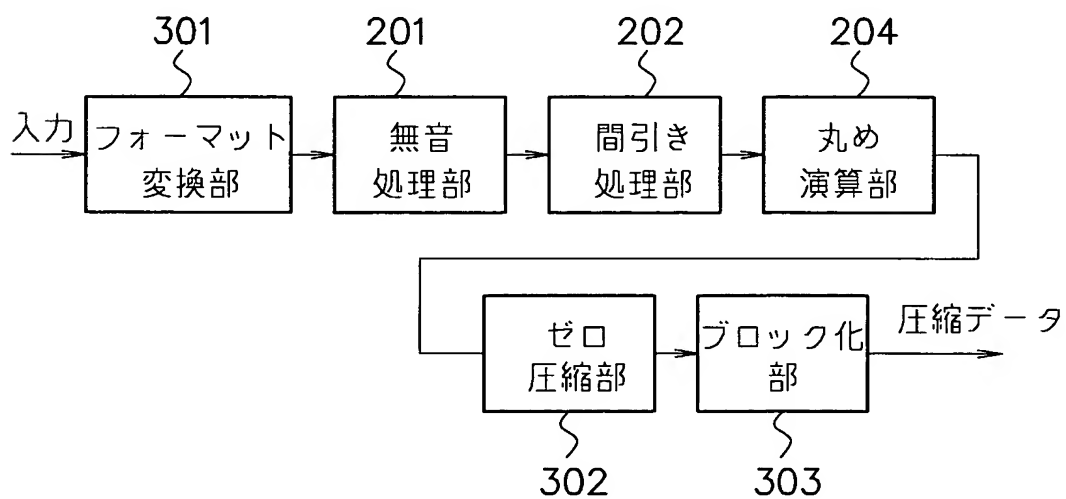


【図 2 4】



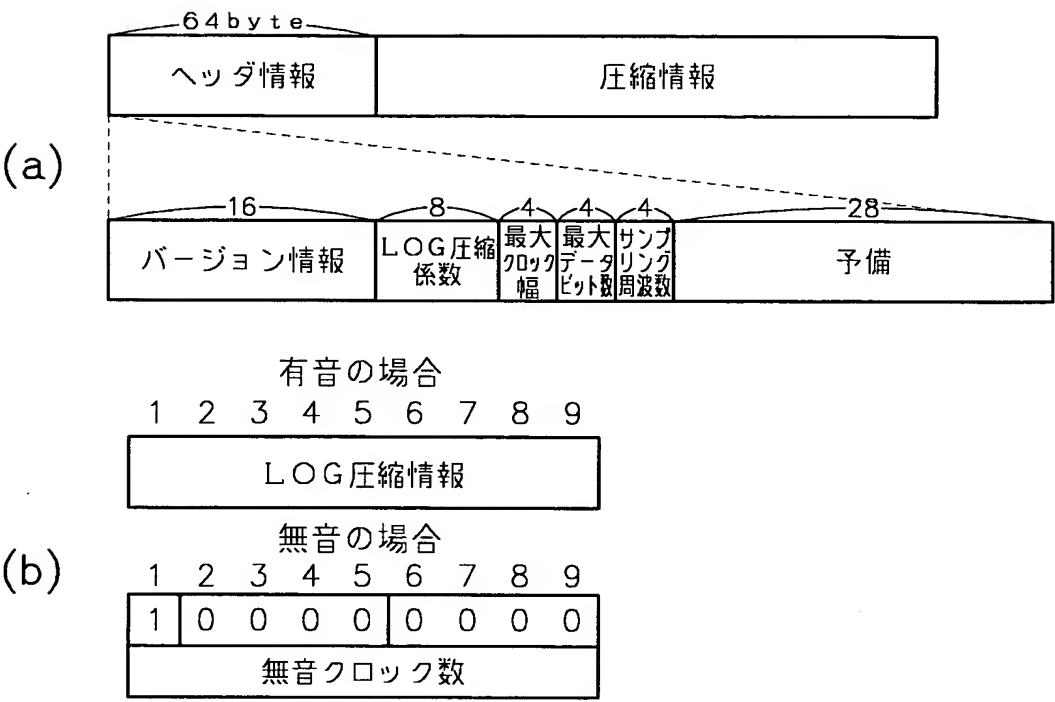
【図 2 5】

## 第 4 の実施形態による圧縮装置



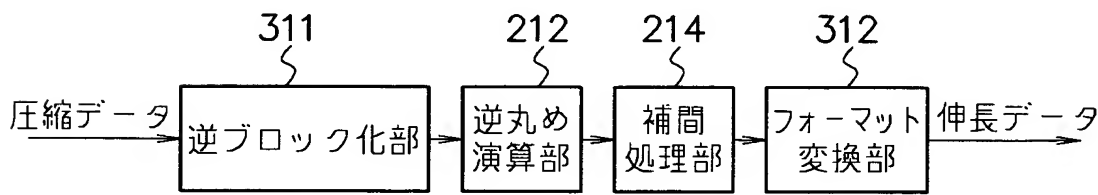
【図 2 6】

圧縮データのフォーマット例



【図 2 7】

第 4 の実施形態による伸長装置



【書類名】 要約書

【要約】

【課題】 圧縮率の向上と再生データの品質向上との両方を実現すること、圧縮・伸長処理を簡素化して処理時間を短くすること、これを実現するための構成を簡単化することが可能な全く新しい圧縮・伸長方式を提供する。

【解決手段】 数段の遅延回路  $1_{-1} \sim 4_{-1}$  と、遅延回路  $1_{-1} \sim 4_{-1}$  の各出力データに対してデジタル基本関数の値に応じた重み付け加算をすることにより、順次入力されるサンプリングデータから間引きデータを求める乗加算回路  $5_{-1} \sim 10_{-1}$  とを設け、全域において 1 回以上微分可能な有限台の標本化関数の元となるデジタル基本関数に基づいて圧縮側で間引きデータを求めることにより、簡単な四則演算のみで最低 8 倍の圧縮率を達成することができるようにする。また、同じデジタル基本関数を用いて伸長側で補間データを求めることにより、簡単な四則演算のみで圧縮前の元データをほぼ忠実に再現することができるようにする。

【選択図】 図 1

【書類名】 出願人名義変更届

【提出日】 平成15年 5月 9日

【あて先】 特許庁長官殿

【事件の表示】

【出願番号】 特願2001-173884

【承継人】

【識別番号】 503084059

【氏名又は名称】 有限会社ニューロソリューション

【承継人代理人】

【識別番号】 100105784

【弁理士】

【氏名又は名称】 橘 和之

【電話番号】 049-249-5122

【手数料の表示】

【予納台帳番号】 070162

【納付金額】 4,200円

【提出物件の目録】

【物件名】 承継人であることを証明する書面 1

【援用の表示】 平成 1 1 年特許願第 1 7 3 2 4 5 号の出願人名義変更届  
に添付のものを援用する。

【物件名】 委任状 1

【援用の表示】 平成 1 1 年特許願第 1 7 3 2 4 5 号の出願人名義変更届  
に添付のものを援用する。

【プルーフの要否】 要

認定・付加情報

特許出願の番号	特願 2 0 0 1 - 1 7 3 8 8 4
受付番号	5 0 3 0 0 7 7 1 4 3 8
書類名	出願人名義変更届
担当官	古田島 千恵子 7 2 8 8
作成日	平成 1 5 年 9 月 8 日

< 認定情報・付加情報 >

【承継人】

【識別番号】	503084059
【住所又は居所】	東京都世田谷区中町五丁目 4 番 7 号
【氏名又は名称】	有限会社ニューロソリューション

【承継人代理人】

【識別番号】	100105784
【住所又は居所】	埼玉県川越市下新河岸 8 7 番地 6 4 橋国際特許事務所
【氏名又は名称】	橘 和之

特願 2 0 0 1 - 1 7 3 8 8 4

出 願 人 履 歴 情 報

識別番号

[ 5 9 5 0 1 6 5 4 3 ]

1. 変更年月日            1 9 9 9 年    5 月 2 4 日  
    [変更理由]            住所変更  
                            住 所        埼玉県浦和市中尾 4 0 9 - 1 - D 1 1 5  
                            氏 名        酒井 康江
  
2. 変更年月日            2 0 0 2 年    3 月    6 日  
    [変更理由]            住所変更  
                            住 所        埼玉県さいたま市中尾 4 0 9 - 1 - D 1 1 5  
                            氏 名        酒井 康江

特願 2 0 0 1 - 1 7 3 8 8 4

出 願 人 履 歴 情 報

識別番号

[ 5 0 3 0 8 4 0 5 9 ]

1. 変更年月日

2 0 0 3 年 3 月 3 日

[変更理由]

新規登録

住 所

東京都世田谷区中町五丁目 4 番 7 号

氏 名

有限会社ニューロソリューション